NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

FEASIBILITY OF AUTOMATING FIWC WEBSITE NONCOMPLIANCE MONITORING AND ENFORCEMENT ACTIVITIES

by

Victoria Josephine Galante

June 2003

Thesis Advisor:	Thomas Otani
Second Reader:	J.D. Fulp

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) **Feasibility of Automating FIWC Website Noncompliance Monitoring and Enforcement Activities**			**5. FUNDING NUMBERS**
6. AUTHOR(S) Victoria J. Galante			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** N/A
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

For written word to reach the public in hardcopy form, a manuscript is submitted to a publisher. After numerous review and modification cycles, the document is printed and distributed, often through intermediaries. Finally, it reaches the hands and eyes of perhaps thousands. This contrasts dramatically with the Internet where, within minutes of completion, text can be seen by millions.

The Internet offers enormous research power. With a PC and a phone line, one can locate a recipe for delicious meringue or deadly ricin; can research a thesis or the step-by-step fabrication of a thermonuclear device. Recognizing the potential for misuse as well as for informing the public, the Department of Defense charged each of its agencies with the responsibility of policing content and form of that agency's publicly accessible websites. As the United States Navy command responsible for this daunting assignment, FIWC faces a job that grows in complexity and size by the day. Taking on this problem manually would result, at best, in unitary growth of dedicated resources and a similar increase in potential for error, both of oversight and of inappropriate action.

This thesis provides one approach to automating FIWC's website monitoring and enforcement activities. The approach it advocates is focused on reducing manpower and increasing accuracy. This architecture – a generic model with a GUI database frontend – is presented, not as an ultimate solution, but rather as a solid first step.

14. SUBJECT TERMS FIWC, Website Monitoring, Record-keeping, Noncompliance, Content Violation			**15. NUMBER OF PAGES** 129
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

FEASIBILITY OF AUTOMATING FIWC WEBSITE NONCOMPLIANCE MONITORING AND ENFORCEMENT ACTIVITIES

Victoria J. Galante
Civilian, Department of Defense
B.S., University of Southern California, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 2003

Author: Victoria Josephine Galante

Approved by: Thomas Otani
Thesis Advisor

J.D. Fulp
Second Reader

Peter Denning
Chairman, Department of Computer Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

For written word to reach the public in hardcopy form, a manuscript is submitted to a publisher. After numerous review and modification cycles, the document is printed and distributed, often through intermediaries. Finally, it reaches the hands and eyes of perhaps thousands. This contrasts dramatically with the Internet where, within minutes of completion, text can be seen by millions.

The Internet offers enormous research power. With a PC and a phone line, one can locate a recipe for delicious meringue or deadly ricin; can research a thesis or the step-by-step fabrication of a thermonuclear device. Recognizing the potential for misuse as well as for informing the public, the Department of Defense charged each of its agencies with the responsibility of policing content and form of that agency's publicly accessible websites. As the United States Navy command responsible for this daunting assignment, FIWC faces a job that grows in complexity and size by the day. Taking on this problem manually would result, at best, in unitary growth of dedicated resources and a similar growth in potential for error, both of oversight and of inappropriate action.

This thesis provides one approach to automating FIWC's website monitoring and enforcement activities. The approach it advocates is focused on reducing manpower and increasing accuracy. This architecture – a generic model with a GUI database frontend – is presented, not as an ultimate solution, but rather as a solid first step.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author thanks all who contributed to the development and documentation of this thesis as well as to the design and implementation of the prototype application. A special thanks to:

The National Science Foundation, for providing the Scholarship for Service Program and for giving me the opportunity to realize a dream;

Dr. Cynthia Irvine, for thinking of me when she first learned of this research and development opportunity;

LT Lucianna Sung, USN (FIWC) for her help and support in providing background material on the thesis subject;

Dr. Thomas Otani, for his expert guidance, his insights, his infinite patience, and for being my unwavering compass;

Mr. J.D. Fulp, for his early involvement, for his encouragement, and for his indispensable contributions – particularly in matters involving naval operations and organization;

Paul Attryde, Robert Healy, and Ben Brady, for their advice and thoughts on Clarion file encryption options;

Ron Jolda, for advice on Clarion tree structures;

Greg Scales, for his comprehensive advice and insight on Clarion tree structures;

Last, but by no means least, I express my profound gratitude to LT Andrew Lamorie, USN (FIWC) without whose prompt, eager, and scrupulously thorough assistance, this thesis and the prototype solution would not have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

I – INTRODUCTION

In late 1998, the Department of Defense – recognizing the potential for adversity in the unregulated publishing of Internet-accessible military information – issued a Policies and Procedures document with respect to the operation and maintenance of all unclassified websites by entities within its domain.

In June 2001, the then-recently formed Naval Fleet Warfare Information Center (FIWC) dispatched a message to all Naval commands, conveying FIWC's official responsibility for oversight and enforcement of the Department of Defense Policies and Procedures regulations within the United States Navy.

Now, just two years later, the explosive growth of the Internet finds FIWC increasingly challenged with the task of staffing and supporting its website oversight and enforcement responsibilities. The increasing use of the Internet by the armed services for public information and liaison bodes no diminution in this trend for the foreseeable future.

This thesis offers a design which automates certain of FIWC's website oversight and enforcement activities. Immediately following this introduction comes the Background chapter, providing specific information on FIWC, the regulations it oversees and enforces, collateral responsibilities, and the six activities whose automation is the focus of this thesis. The next chapter, entitled Architecture Design, documents the design objectives, supporting functionality, security considerations, database design, and provides a brief overview of the prototype application. This overview is amplified in the following Implementation chapter. Here, the prototype's implementation, development methodology, and the most significant of its sixty-four procedures are presented. The Implementation chapter goes on to discuss features which, although either partially implemented or unimplemented in the prototype, should be considered for incorporation into a production-class system. Finally, the Conclusion recounts how the project evolved from our initial vague perceptions, through false starts, and finally into a coherent research document and a concise prototype implementation. The Conclusion then

recapitulates the unimplemented features described in detail in the Implementation chapter. In this recapitulation, the author provides a manpower estimate for completion of each of the thirteen unimplemented features.

It is the author's sincere hope that some part of the research and development effort presented in this document will be of assistance to FIWC in the conduct of its crucial mission.

II – BACKGROUND

On 25 November, 1998, the Department of Defense published a Policies and Procedures document entitled, "Web Site Administration Guidance." This document sets forth policy and responsibilities related to the operation and maintenance of all unclassified websites – regardless of whether publicly accessible – by agencies and departments within the scope of authority of the Department of Defense.

Section 5.5 of this policy states that "Heads of the DoD Components[1]" have the following responsibilities:

1. Establish procedures for identifying website-appropriate information and ensuring that the procedures are consistently applied;
2. Ensure that all information placed on publicly accessible websites is properly reviewed for security, levels of sensitivity and other concerns prior to release;
3. Ensure that approved, DoD security and privacy notices and applicable disclaimers are displayed on all websites;
4. Ensure that all information placed on publicly accessible websites is appropriate for worldwide dissemination and does not compromise national security, DoD personnel and assets, mission effectiveness, or individual privacy;
5. Ensure that procedures exist for websites management oversight and regular functional review;
6. Ensure the operational integrity and security of all website-supporting computers and networks;
7. Ensure reasonable efforts to verify the accuracy, consistency, appropriateness, and timeliness of all information placed on websites;

[1] For full text of the policy, amended as of 11 January 2002, please refer to Appendix A

8. Register publicly accessible websites with NWRS;
9. Provide adequate funding, equipping, staffing and training resources to support website operation.
10. Conduct comprehensive, multi-disciplinary, website security assessments no less frequently than once a year;
11. Provide a "Lessons Learned" feedback mechanism for other DoD organizations;
12. Ensure policy compliance for all “functions, missions, agencies, and activities” within their jurisdiction;
13. Grant policy waivers and deferments under certain, specified conditions;

FIWC, the Fleet Information Warfare Center, is the organization within the U.S. Navy which has responsibility for carrying out the Web Site Administration Guidance Section 5.5 directive.

FIWC was established and became operational on October 1, 1995 as the U.S. Navy's "Center of Excellence for Information Operations."

Located at Little Creek Amphibious Base, Virginia Beach, Virginia, FIWC provides Information Operations (IO) support to Naval Forces worldwide. In its just over seven years of existence, FIWC has worked with deploying Fleet staff and Naval units, providing Navy-wide seminal support in Computer Network Defense and Electronic Warfare.

This support includes computer incident response, vulnerability analysis, and incident measurement services, and entails providing facilities, equipment, and personnel for the direction of the defensive information warfare program, including detecting and responding to computer attacks[2]. In March 2000, FIWC was awarded the prestigious Navy Meritorious Unit Commendation[3]. In July 2002, FIWC was placed under the Naval Network Warfare Command (NNWC), or “NETWARCOM.”

[2] FIWC Public Affairs Office (PAO) http://www.fas.org/irp/agency/navsecgru/fiwc/

[3] From the FIWC main website, http://www.fiwc.navy.mil/.

LT Andrew Lamorie asserted: "FIWC is responsible for monitoring all publicly accessible, U.S. Navy NIPRNET/unclassified websites for conformance to various regulations governing registration[4], content, form, and authorization to publish." As of 7 May 2003, approximately 3,200 U.S. Navy websites fell under FIWC's domain.

On 12 June 2001, FIWC dispatched a NAVCIRT/NCTF coordinated general administrative message to all Naval commands, stating[5]:

> … FIWC is responsible for conducting random web site verification checks and providing non compliant commands with specific data concerning non compliance. Previously, the area of operational security (OPSEC) was focused on activities that might only be seen by a human observer, a satellite, news, etc. The newest area of concern and vulnerability is the Internet. In an effort to reduce the amount of sensitive information that is posted on publicly accessible web pages, FIWC was tasked to assess DON web sites for compliance with applicable directives.

After this introduction, the message goes on to list a subset of the website compliance regulations. At the time of this writing, the regulations are covered by the following documents:

- Ref A: DoD Web Site Administration Policies and Procedures (mentioned in preceding paragraphs)[6]
- Ref B: SECNAVINST 5720.47[7]
- Ref C: NAVADMIN 088/99[8]
- Ref D: SECDEF Memo 28DEC2001[9]
- Ref E: SECDEF Memo 13JUL2000[10]

---

[4] An unregistered, U.S. Navy, publicly accessible website, commonly known at FIWC as a "rogue website." Please see Appendix for a sample registration form.

[5] Case changed from all caps to sentence case and one typo fixed for readability.

[6] http://www.defenselink.mil/webmasters/policy/dod_web_policy_12071998_with_amendments_and_corrections.html

[7] http://neds.nebt.daps.mil/Directives/5720_47.pdf

[8] http://www.bupers.navy.mil/navadmin/nav99/nav99088.txt

[9] http://www.defenselink.mil/pubs/foi/names_removal.pdf

[10] http://www.c3i.osd.mil/org/cio/doc/cookies.html

Please see Appendix B for the full original text of the message.

A. FIWC'S WEB MONITORING RESPONSIBILITIES

FIWC's web monitoring responsibility consists of six activities: Assessment and Discovery, Documentation, Review, Record-keeping, Citation, and Verification. The following is a brief description of these six activities:

1. Assessment and Discovery

The Assessment and Discovery activity encompasses the review of all covered websites against regulation-defined compliance criteria. Compliance reviews are performed annually, and are broken into twelve monthly cycles. The current FIWC plan calls for roughly $1/12^{th}$ of the approximately 3,200 sites to be reviewed each month by fourteen reserve units[11]. About 15 URLs are assigned monthly to each of these fourteen units, supporting the current, annual-review target of 2,520 websites.

During the benchmark month of April 2003, the FIWC reserve units reviewed a total of 166 URLs. It is important to note that the assigned units are responsible for searching the web for discovery of unregistered Naval websites in addition to compliance reviews. At the time of this writing, two active duty personnel are assigned to the unregistered Naval websites search, in addition to other tasks.

The purpose of the assessment and discovery activity is to locate unregistered sites and sites otherwise in violation of the governing regulations. Once a site is found to be in violation, it progresses to the next monitoring phase.

2. Documentation

This includes registration of Naval websites, recording of URLs containing violations[12], violation particulars, responsible commands and other information pertaining

[11] The fourteen reserve units' sizes vary from one to nineteen assigned personnel, for a total of approximately 85 people. Each reserve unit drills approximately three out of four weekends a month, but any given reservist only drills one weekend a month. This is the reason given for performing assessment on monthly cycles.

[12] A violation is also known within FIWC as a "discrepancy."

to websites found to be unregistered[13] or in violation of statutory regulations. Multiple violations on a webpage are separately entered and accounted for. Each of these violations, if confirmed, is recorded as a separate, citable incident. All site URLs in violation are linked to the URL of the home page.

3. Review

Once a website, or any publicly accessible web page subordinate to a website, is determined to be in violation, the documentation produced by the reservists is subjected to a QA (quality assurance) review. The primary purpose of this review is to either confirm or invalidate the preliminary violation assessment. Secondary purposes of the QA review are to ensure that a high-quality assessment is sent to the webmasters and that no major violations are overlooked.

4. Record-keeping

This covers the journaling of confirmed violations, recording remedial actions, issuance and recording of notifications of violation and grace period, issuance and recording of directed Navy messages (hereinafter, simply "directed messages"), updating registrations, and entering registrations into NWRS[14]. Websites and commands are tracked using the home page URL. A special, automatic procedure will perform a daily search of the WebRAT[15] database for cited websites with expired discrepancy report grace periods. This procedure is defined in a slightly different manner in the proposed implementation (please refer to Chapter IV, Implementation, for additional detail).

5. Citation

Citation consists of two sub-activities:

1. An informal e-mail contact (i.e., "Memorandum of Violation and Grace Period") by FIWC to the violating site's webmaster, wherein the

[13] Naval regulations require that websites register in NWRS (Naval Website Registration System). Registration enumerates command particulars. Active duty staff manually verify and enter these registrations into the Microsoft Access-based NWRS database.

[14] Dissatisfied with GILS, FIWC created its own database to support registration of publicly accessible, Navy web sites. Before FIWC's review, there were approximately 1300 registered Navy web sites. By the time FIWC had finished the review, this number had grown to 3175 registered Navy web sites, of which 460 were listed as dormant.

[15] WbeRAT: Web Risk Assessment Team

webmaster is advised of the violation particulars and is provided a copy of the assessment report. A grace period of 30 days is granted for the correction of violations.

2. FIWC issues a directed message to any command with website violations unremedied after expiration of the 30-day grace period. The directed message is issued Naval Component Task Force (NCTF), who will then send "record message traffic" to the website's Echelon II commander, reporting the URL's command for failure to correct noted discrepancies. Echelon II commanders are three- or four-star admirals with responsibility for geographic or operational areas of command. There may be zero or more commands intervening between the website command and the Echelon II command.

It is important to note that certain levels of violation severity (e.g., the publishing of military-sensitive information) may cause FIWC to take immediate action to either: a) silence the violation or b) shut down the site, thus contravening the normal procedure of notification and grace period.

6. Verification

Webmasters are required to report compliance of the website within thirty days of receiving notification. Once an officially cited command's webmaster has notified FIWC of remedy, FIWC staff verifies full compliance by conducting a comprehensive site verification reassessment.

The following is an abbreviated list of specific violations of the five regulations cited earlier in this chapter. Violations deemed severe in the author's opinion are highlighted by underline. This emphasis is that of the thesis author and in no way is intended to reflect official, DoN position. A complete list of violations can be found in the Appendix E:

<u>Omission</u>

- Insufficient Rank for publicly accessible web site
- Failure to contain full command's name

- Failure to state “This is an official U.S. Navy web site”
- Failure to provide standard Privacy and Security Notice
- Failure to provide webmaster contact information
- Failure to provide a link to parent command or Immediate Superior in Chain (ISIC)
- Failure to provide a link to the official U.S. Navy web site
- Failure to provide a link to the Navy recruiting web site
- Failure to provide disclaimers on links to other than U.S. Government web sites
- Failure to provide Privacy Advisories on all site visitor solicitations
- Failure to have written SECDEF approval for persistent cookies
- Failure to provide a disclosure for all session and approved, persistent cookies
- Failure to provide Notice & Consent (DoD Warning) Banner at an access point controlled by level-3 security (Authentication)

Commission

- Presence of any warning with respect to the Privacy and Security Notice
- Certain photographic alterations
- FOUO or above information
- Personally identifying content (e.g., social security number, marital status, age, home address, home phone number, birthdate, place of birth, family members, race, religion, citizenship, city home of record, personalized email address)
- Proprietary or copyrighted content
- Operational Lessons Learned

- Information on sensitive military operations, exercises, vulnerabilities, maps, etc.
- Specialized, internal information or information of questionable value to the general public
- Information that places national security, personnel, assets, or mission effectiveness at unacceptable risk
- Phone numbers that can be associated with individuals
- Product endorsements, preferential treatment of any private organization or product, or references including logo or text indicating that the site is "best viewed" with any specific web browsers
- Contain links or references to documents within DoD Web sites that have security and access controls
- Content duplicated from other military web resources

B. MANPOWER

FIWC estimates[16] that it expends about 980 person hours per month on the six activities enumerated previously. This estimate is distributed as shown in Table 1 - Manpower Distribution

.

[16] Estimates provided by LT Andrew Lamorie

Activity	Person Hours
Discovery	700 +
Documentation	60
Review	160
Record-keeping	40
Citation[17]	0
Verification	20
Total:	980 +

Table 1 - Manpower Distribution

C. PROBLEMS TO BE REDUCED OR ELIMINATED

When asked what problems FIWC hoped to reduce or eliminate through automation, the following, paraphrased response was given: *The Discrepancy Tracking database called 'Remedy' is cumbersome and labor intensive to use and does not work with the MS Access NWRS database. For example, five discrepancies on one URL require five distinctive database entries.*

FIWC presently handles an estimated 1.3 terabytes of web-published, Naval material, and is observing exponential growth. Extending this trend, FIWC estimates that, lacking more efficient procedures, web risk assessment (WRA) will ultimately require over 500 full-time personnel. Currently, the Navy employs 115 part-time people for WRA, and has neither plan nor budget to expand staffing for this function.

D. CURRENT ACTIONS AND PLANS

FIWC is developing a parser program to automatically log violations and associated URLs into a database. QA personnel will then review the violations and will confirm the assessment. Once the assessment is confirmed, FIWC staff will copy it to a report to send to the webmaster. If the assessment fails confirmation, they will invalidate the assessment. Invalidated assessments will be deleted from the database. It is possible for additional violations to be discovered during the QA process.

FIWC maintains close communication with other branches of the DoD, the U.S. Coast Guard, and JWRAC (Joint Web Risk Assessment Cell) with respect to the DoD

[17] Considered negligible

Web Site Administration Guidance directive. Furthermore, FIWC regularly supplies VADM Mayo of NETWARCOM with WRA (web-risk assessment) information. The U.S. Coast Guard and U.S. Marines are working with FIWC to adopt WRA policies and procedures similar to those implemented and under design at FIWC.

As is presently practiced, an informal email memorandum of violation will be sent to the webmaster. It is planned that automatically generated reports will identify websites that have passed the grace period or that are due for annual assessment. These reports will be run daily.

E. OTHER SOLUTIONS IN PLACE

All agencies and departments within scope of authority of the 25 November 1998 Department of Defense Policies and Procedures Web Site Administration Guidance directive face a similar challenge. In that regard, as mentioned in the preceding paragraphs, FIWC has kept close ties with other DoD agencies who are engaged in similar activities. Please refer to Chapter II – Background, for additional information on the DoD Web Site Administration Guidance directive.

Both the U.S. Navy (FIWC) and the U.S. Air force have licensed the COAST's Web Quality Central product. Web Quality Central provides automated website compliance review and accounting functionality. JWRAC (DoD's Joint Web Risk Assessment Cell) is presently reviewing an evaluation copy of Web Quality Central. It is FIWC's intent to couple the Web Quality Central site analyzer with GOOGLE's search engine to capture domain web content and to feed the derived information to WebRAT. This automated search and retrieval system will seek out what FIWC terms "low hanging fruit," the easily detected violations. FIWC expects that the automated solution, called "WebRAMMS," will continue to be supplemented with personal site evaluations by FIWC staff. In addition to seeking and discovering violations, website registration information is passed to NWRS, the Navy Website Registration System.

F. PURPOSE OF THESIS

The main purpose of this thesis is to provide a research and implementation guide for automated support of FIWC's Assessment and Discovery, Documentation, Review, Record-keeping, Citation, and Verification activities.

It is hoped that implementation of the prototype documented herein will enhance FIWC's execution of these activities by:

Making more efficient use of manpower;

Reducing typographical and classification data entry errors;

Increasing accuracy and confidence over issuance of memoranda of violation and directed messages;

Improving speed and accuracy of locating site and command violation information;

Augmenting the timely access of historical violation information, both for online-view and hardcopy report production.

This thesis provides a detailed examination of the issues surrounding the automation of these six activities, a specification of the automated solution, and a working prototype of that solution.

THIS PAGE INTENTIONALLY LEFT BLANK

III – ARCHITECTURE DESIGN

The following is a list of design objectives for the *FIWC Website Compliance Database and Application* (hereinafter "application"):

- The application must support and facilitate FIWC's assessment, discovery, documentation, review record-keeping, citation, and verification functions[18].
- The application must have a friendly and flexible user interface.
- Due to the severity of non-compliance action, the likelihood of error – both of commission and of omission – must be reduced to the greatest feasible extent.
- The application must be easily modified and readily adaptable to FIWC's growing needs and changing demands.
- The application and database must maintain an appropriate level of security and integrity.

These objectives are corroborated by the following functionality:

- Minimization of record-keeping workload;
- Automated generation of informal memoranda of violation and directed messages[19];
- Reduction of data entry errors;
- A graphical user interface with maximum use of drop lists, multi-key browse forms, and multi-sequence reports;
- Reasonable and appropriate security measures.

A. REASONABLE AND APPROPRIATE SECURITY MEASURES

There are several issues which bear on *FIWC Website Compliance Database and Application* security. Where applicable, the prototype implementation described in Chapter IV – Implementation, addresses these issues.

[18] Please see Chapter II – Background, for more information on these activities

[19] Notifications of violation and directed messages are only partially implemented in the prototype. However, implementation of both is supported by the Clarion development platform.

1. Database Security

Some of the violations in the database may be of a sensitive nature (e.g., FOUO[20] information; "Information that places national security, personnel, assets, or mission effectiveness at unacceptable risk [Ref A, part II, 3.6.2, part V, 2; Ref B, encl 2: 3.d.1]").

While it might require substantial time and effort to locate such compromising information on the Internet, the *FIWC Website Compliance Database* would provide a convenient index to such material. Although physical security of storage media and computer/s with access to the database are beyond the scope of this thesis, the software architecture, policies, and procedures are not and are therefore addressed in the following paragraphs.

2. Accessibility

The *FIWC Website Compliance Application* is the main channel of access to the database. The application must provide a secure portal, requiring authorized password for admission to the system. The password table should be encrypted (as in the prototype implementation). Database administrative personnel should ensure that user-selected or user-devised passwords are hack resistant (e.g., ten or more characters in length, lacking verbal significance, and containing at least one case-variant letter, one numeric character, and one special character).

3. Unscheduled Review

An audit review of the database should be performed periodically, on an irregular and unpublished schedule, by supervisory personnel to search for irregularities.

4. Database Exposure

Imposing the privilege of least privilege, exposure of the database should be as restrictive as is consistent with operational requirements. E.g., barring compelling reason to the contrary, the application and database should not be Internet-accessible.

5. Backups

The *FIWC Website Compliance Database* should be backed up regularly. A recommended practice is full or at least partial backups daily, full backups weekly, and an offsite archive backup monthly. Offsite archive retention should be consistent with practices for materials of similar sensitivity and import.

[20] For official use only.

6. Assignment Rotation

Website assignment should be rotated periodically among FIWC staff.

7. Separation of Duties

Different personnel should be assigned responsibility for locating, recording, and taking action on violations.

8. Encryption

If the database is readily accessible by means other than the *FIWC Website Compliance Application* (e.g., SQL or Access inquiry and/or update), the *Violations* table should be considered for encryption. SoftVelocity, owner of Clarion, provides a proprietary table-encryption algorithm for the TopSpeed driver. This algorithm is employed for encrypting the Prototype *Chains* (password) table in the prototype implementation. This is a proprietary encryption algorithm, neither documented nor identified by SoftVelocity.

A third-party developer, Brady and Associates, LLC[21], also provides a common MD5 encryption capability. In addition, it is rumored that another third-party developer is implementing Blowfish encryption for Clarion, but no details of this implementation are known to the author.

9. Audit Trail

An external[22] audit trail capable of posting critical updates should be considered. Critical updates should be selectable by supervisory personnel on the fly, and would key on table, action and attribute.

For example, an audit trail could be triggered on all insert, update, and delete changes affecting the *Status* (*Vid:Status*) and *Disposition* (*Vid:Disposition*) attributes in the *Violations* table. Information logged should be:

- Table name;
- Attribute name;
- Date and time of access;

[21] http://www.clariondeveloper.com

[22] "External audit trail," in this context means a procedure embedded in the application which surreptitiously logs auditable events to a table external to the application database.

- Nature of access (i.e., Create, Update, Delete);
- Pre- and post-modification values;
- Identity of the person making the update (as obtained from the global item *Glo:StaffId*, captured at login time).

An unreviewed audit trail is worse than useless, because logging audit events consumes substantial CPU time and disk space. The audit trail, if enabled, must be reviewed, and apparent anomalies must be rigorously investigated and acted upon. Only if such a level of commitment is forthcoming, should an audit trail be implemented.

10. Database Segmentation

Database segmentation, as described in Chapter IV – Implementation, would confine *Violations* access to the person to whom the violations were assigned. Segmentation reduces the likelihood that someone other than the assigned staff person can illegally delete or place an unauthorized value on a violation. It also addresses the grim possibility of a webmaster and FIWC staff personnel colluding to advertise sensitive information on an obscure domain link.

11. Mistakes

More bad consequences flow from honest error than from intentional malfeasance. Errors, both of omission and of commission, in the *FIWC Website Compliance Application* can have adverse consequences:

- Delayed Action: There is a broad severity range for website violations, ranging from mild (e.g., failure to include a required link, statement, or heading) to grave (e.g., publishing information which could compromise national security). It is important that high-severity violations be acted on in a timely manner. The system designer should consider a special proactive procedure for inclusion within the application. This procedure would initiate popup alerts on violations exceeding a specified severity threshold.
- False Positives: A DNM or directed Navy message carries serious consequences, both for the command and for the admiral in charge. The application must include all reasonable safeguards to prevent the creation and dispatch of an erroneous directed message.

- False Negatives: A crucial violation, incorrectly flagged as cited or unfounded, essentially defeats FIWC's charter. Audit trails and/or requiring supervisory privilege to change a violation's status or disposition would mitigate this risk.

B. DATABASE DESIGN – THE PROCESS

Because the record-keeping phase of FIWC Website Compliance Application is data-centric by its nature, database design was viewed as essential to a well-formed and flexible application. The database design steps were:

1. Identify and briefly describe data attributes;
2. Publish interim document, seek feedback from FIWC and thesis advisor, and modify accordingly;
3. Seek feedback from other I.T. professionals with respect to complex, Clarion- or TopSpeed-specific areas of the design (e.g., data encryption, navigating the chain-of-command tree);
4. Group the reviewed and modified, semi-final set of attributes into tables;
5. Publish interim document, seek feedback from FIWC and thesis advisor, and modify accordingly;
6. Define tables and attributes in a schema;
7. Define all inter-table relationships within the schema;
8. Specify relational-integrity constraints on table relations;
9. Designate reference-table validation within the schema;
10. Complete dictionary specification of tables and (especially) attributes[23];
11. Select a prototype DBMS capable of supporting the defined data structures and relationships;

[23] The Clarion dictionary supports extensive attribute characterization, including but not limited to: formal identifier, prompt identifier, column identifier, data type, length, scale, default form control, domain and range validation, display editing, dimension (i.e., array), prompts, case, initialized values, and flags (e.g., "password").

12. Generate the dictionary and application framework.

Once these steps were completed, detailed application design and then implementation could (and did) commence.

C. DATABASE DESIGN – SPECIFICS

The FIWC Website Compliance Database was designed under the relational model. The relational model is not only well tested and widely used, but also accommodates the prototype design goals. Object models, i.e., OODBMS and OORDBMS, were not considered, not because of lack of applicability but because of insufficient resources.

Many relational and object database management systems would have supported this prototype. Of the dozen relational database management systems (and nine reasonable choices) supported by the Clarion 4a development platform, we chose TopSpeed[24] for its speed and the robustness of its implementation.

TopSpeed is a full, relational DBMS, and provides all of the features we deemed essential to the implementation. Here is a list of the dozen DBMS's considered[25] for the application:

1. ASCII (primitive),
2. Basic (primitive),
3. Btrieve,
4. Clarion,
5. Clipper,
6. dBase-III,
7. dBase-IV,
8. DOS (primitive),

[24] A complete list of TopSpeed DBMS features is presented in the Appendix.

[25] In addition to these, for which Clarion has native drivers, Clarion also supports ODBC interface for SQL and other unlisted RDBMS.

9. FoxPro,
10. SQL (via ODBC interface),
11. Access, and
12. TopSpeed.

The following discussion describes the database in general, functional terms and provides a brief description of each table. Please refer to Appendix F for a comprehensive list of TopSpeed database driver features. Attribute and table names are presented in rhetorical format in the body of the thesis for readability. For example, the table "ParentCommands" is referred to here as *Parent Commands*. Table and attribute names are presented in italics.

The FIWC Website Compliance Database consists of twelve tables[26], each of which falls into one of three classifications. The tables are distributed as follows:

- One administrative table,
- Three primary tables, and
- Eight reference tables.

The FIWC Website Compliance Database ER Diagram illustrates the database tables and their relationships.

Chains is the single administrative table. In fact, *Chains* contains one tuple[27] for every *Staff Id* registered for system access. The remaining *Chains* attributes are *Password* and *Privilege Level*. *Password* accommodates strings of up to 20 characters, each of which may contain any upper- or lower-case alphabetic character, the digits 0 through 9, and any special ASCII character (e.g.,!, @, #, $, etc.). The *Chains* table is encrypted[28] and comes primed with two *Chains* tuples, one possessing a Supervisory privilege level.[29]

[26] "Table" as used here is synonymous with "file" and "dataset."

[27] "Tuple" as used here is synonymous with "record" and "entry."

[28] The unpublished encryption algorithm is proprietary to SoftVelocity, owner of Clarion.

[29] The User Id (StaffId) and Password for supervisory access are "1" and "V1ck1eJ0" respectively [Note upshifted initials and the digits one and zero in the password.]. Password is case sensitive. Long (ten-character or longer) passwords containing letters of varying case, digits, and special characters are recommended.

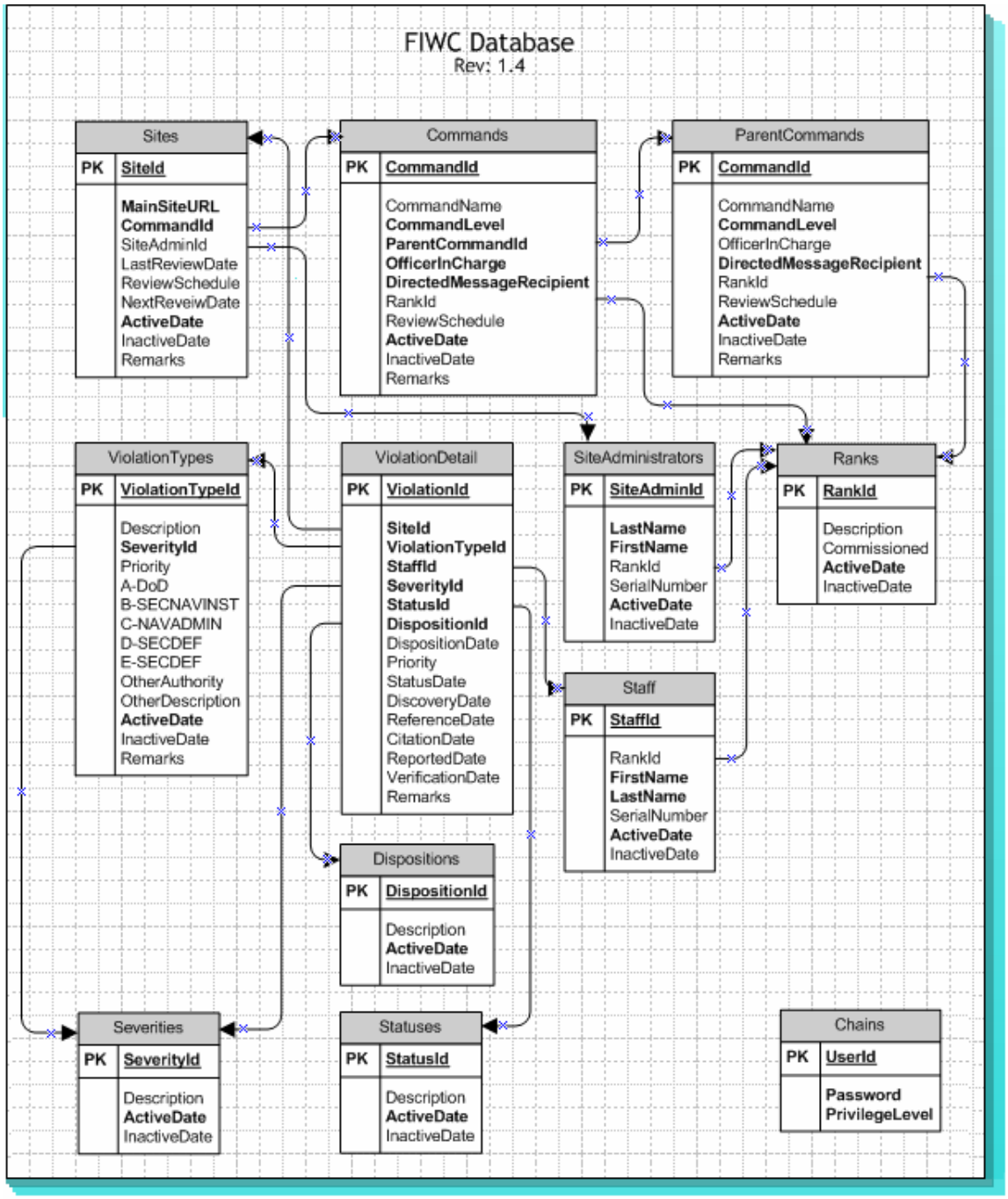


Figure 1 - FIWC Website Compliance Database ER Diagram

The primary tables are:

Commands – There should be one tuple for every command in the FIWC domain. This includes commands both up- and down-stream from a citable command, i.e., one designated as a directed message recipient.

Parent Commands – Parent Commands is used to provide a tree structure, which emulates the Commands' organizational structure. It is a virtual duplicate of the *Commands* table, with a few minor attribute variances. Furthermore, leaf-node commands (i.e., commands without subordinate commands) need not be present in *Parent Commands*.

Violations – The information heart of the database, *Violations* contains a tuple for every discovered violation, whether cited or subsequently determined to be unfounded. Each *Violations* tuple contains a complete history of the violation, including a memo area for freeform notation.

Eight reference tables directly or indirectly support the primary tables. Their common purposes are:

- To increase accuracy by reducing case, spelling, and other typographic and idiosyncratic discrepancies;
- To reduce the time spent entering recurring information;
- To maximize search and sort coherence by reducing the incidence of superfluous search and sort key synonyms[30];
- To provide consistency in online views and reports.

Although reference tables, range checks and other validation devices help to increase accuracy, they cannot eliminate user error. For example, a user could create a spurious reference table entry by adding the practically synonymous Rank reference tuple "Adm." in addition to the legitimate "Admiral." The user can also choose an incorrect reference tuple, e.g., the disposition "Pending review," instead of "Cited" for a Violations tuple, where "Cited" is the correct choice.

Regardless of the elegance and ingenuity of a system's design, the user ultimately holds the key to database accuracy and congruency. The most primitive, manually posted spreadsheet kept by a meticulous clerk is preferable to a state-of-the-art, automated solution maintained without discipline.

[30] It is not possible for reference tables to completely eliminate synonyms, which may be legitimized by inclusion in the reference set.

The reference tables are:

Ranks – Supports *Parent Commands*, *Commands*, *Site Administrators*, and *Staff*;

Sites – Supports *Violations*;

Site Administrators – Supports *Sites*;

Staff – Supports *Violations* and *Chains*;

Dispositions – Supports *Violations*;

Severities – Supports both *Violations* and *Violation Types* (*Violation Types* provides an optional, default *Severity* value when linked to a *Violations* tuple);

Statuses – Supports *Violations*;

Violation Types – Supports *Violations*.

The Clarion schema and dictionary support table relationships, whose relational integrity constraints are both integral and automatically enforced. For example, the application will not permit deletion of a tuple where such action would create an orphan. Specifically, no reference tuple with a live primary link (i.e., a primary tuple to which it points) can be deleted.

There are two static lists (exclusive selection or "radio buttons" for the *Chains* attribute *Privilege Level*, and inclusive selection or "check boxes" for governing regulations in *Violation Types*) in the database. However, all dynamic but finite lists are expressed as primary reference table relationships.

D. APPLICATION OVERVIEW – HOW THE PIECES FIT

There is an automatic sequence of events when the application is dispatched:

1. The main MDI procedure receives control. The MDI procedure immediately invokes both…
2. The login form and…
3. A special FIWC splash panel, which displays for five seconds or until the user clicks within the splash borders, whichever occurs first.
4. Upon successful login, the MDI window form becomes active.

5. The Violations Browse thread is invoked automatically by the MDI procedure, and the Violations Browse form displays. [Note: This option is disabled in the prototype implementation.]

6. The application is ready for use.

The following figure diagrams this set of relationships.

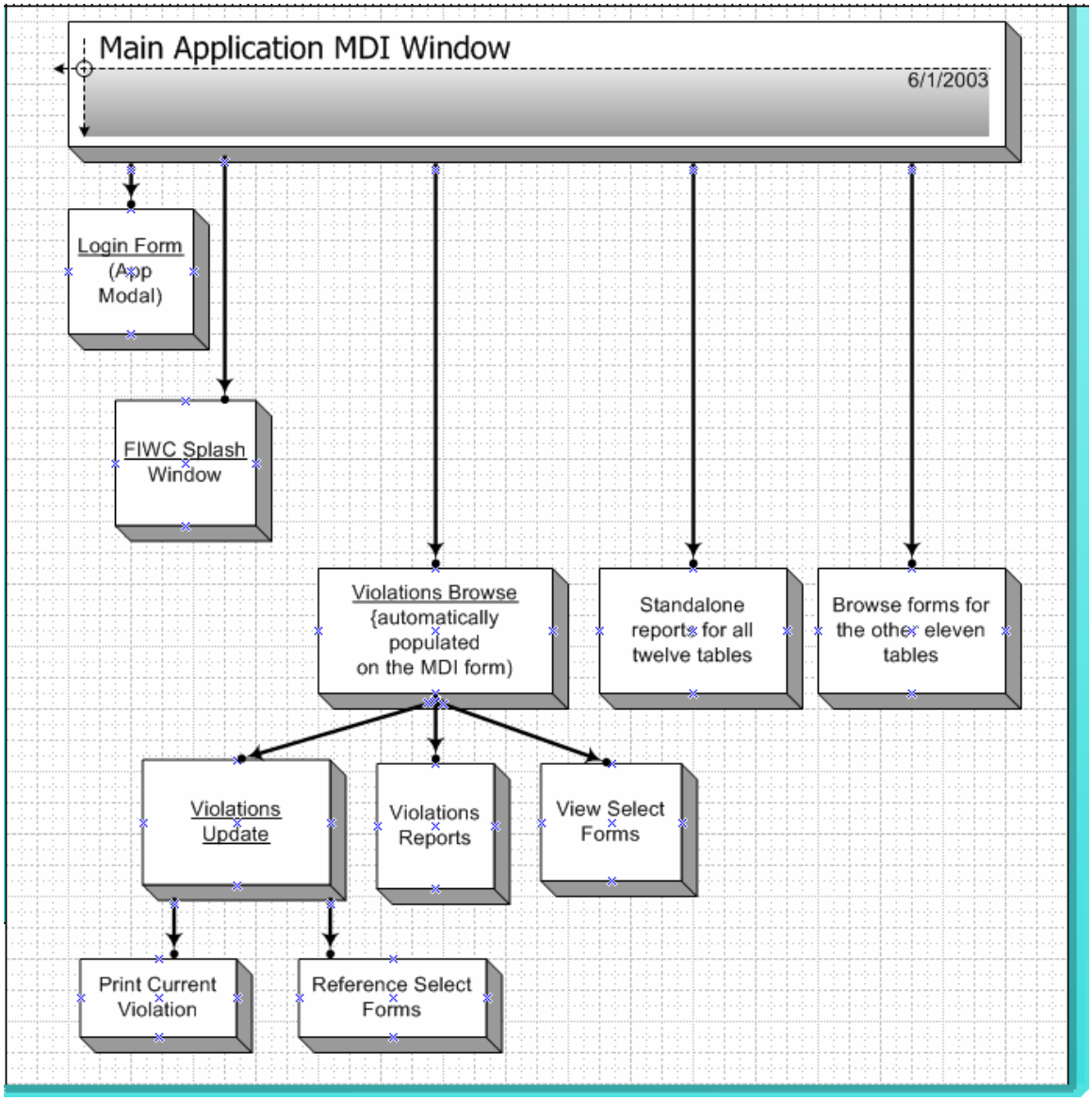


Figure 2 - The Main MDI Procedure

From here, the user can close the Violations Browse or, more likely, work with it. Because browses and updates are unique threads, the user can invoke other browses and reports from the menu bar before terminating the current browse or update procedure.

From each browse form, the user can insert, change, and delete individual tuples in addition to selecting browse views (tabs) and reports. The other eleven browse procedures support various combinations of these features, as appropriate.

IV – IMPLEMENTATION

This thesis includes the implementation of a prototype, which is offered as proof of concept. This implementation has two components:

1. A database
2. A GUI application

Database architecture is covered in depth in Chapter III, Architecture Design.

The prototype application was implemented both to lend substantive support to the assumptions and constraints cited in chapter III, and to provide an alternative solution design by someone external to the FIWC organization. Although formally designated as *FIWC Website Compliance Database* and *FIWC Website Compliance Application*, the following discussion simply uses the words database and application for the most part. Furthermore, the word "application" may be construed here to include the database as well, depending on context.

A. DEVELOPMENT METHODOLOGY

Of the several implementation tools available (including third- and fourth-generation languages), we decided on a RAD development tool. The reasons for this decision are explained in the following rationale. First, for those unfamiliar with this development platform, here is a brief introduction.

RAD (Rapid Application Development) tools include such commercial off-the-shelf products as PowerBuilder™, Clarion™, C++Builder™, and Delphi™. RAD tools provide rich implementation functionality. For example, where third- and fourth-generation language IDEs[31] may offer only a semantic palette and pre-defined classes, RAD tools provide those plus a visual design environment for creating seamlessly integrated GUI forms, controls, and templates. Although all RAD tools offer the advantages of fast and extensible prototyping, each has its peculiar strengths. Clarion was selected for the following reasons:

[31] IDE: Integrated Development Environment

1. Datacentric — Although Delphi and C++Builder both accommodate the integration of a database into a developed application, the binding between database schema and application is not tight. With Clarion, however, the database can be defined as the cornerstone of the nascent application. Data structures are the very foundation on which the data-centric Clarion application is built. Although a Clarion application can he designed procedurally, the preferred and common first step for any Clarion database application is the definition of a comprehensive schema and dictionary.

During schema and dictionary design, tables, attributes, primary and foreign keys, data types, scaling, referential integrity constraints, and default control formats (in addition to numerous other specifications) are specified. A thoughtful and thorough data definition is essential to the development of an optimal application. Because the FIWC record-keeping function is datacentric, a datacentric RAD tool was the natural choice for a development platform.

2. Multi-Level Implementation — Clarion accommodates project and application modification at the following levels:

1. Dictionary,
2. GUI,
3. Code, and
4. Template.

Modifications at all four levels are supported concurrently and non-preemptively. In other words, the developer can effect change at the level most appropriate to the task at hand, easily hopping from one level to the next, without restriction or loss of code. And, although quite different in development "look and feel," modifications at all levels are transparently integrated when the application is compiled. There is no penalty for choosing the most appropriate modification archetype for the task at hand.

3. Comfortable and Intuitive User Interface — Updates, inquiries, and reports are executed in a user-friendly and intuitive GUI interface. For example, the Violations browse form offers the user seven views of the Violations table:

- Violation order,

- By selected Site,
- By selected Violation Type,
- By selected Staff Person Assigned,
- By selected Severity,
- By selected Status, and
- By selected Disposition.

Each view, selected by clicking on the appropriate tab, gives the user an appropriate focal category of interest (e.g., Site, Severity, etc.).

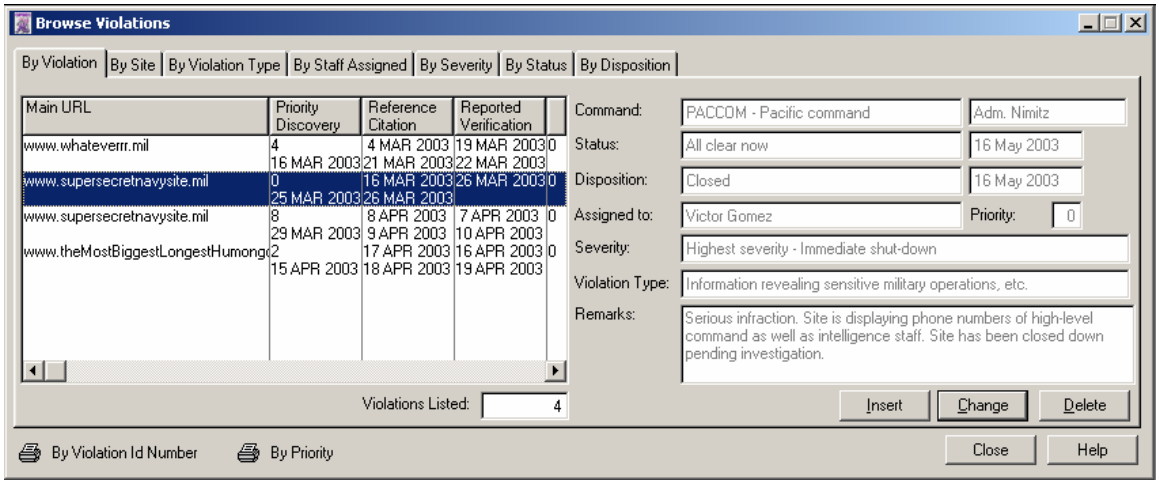


Figure 3 - Violation Browse

Each of the seven views provides click-of-the-button reporting, with reported violations clustered according to the tab-indicated criterion.

Clicking either the Insert or the Change button invokes the update form.

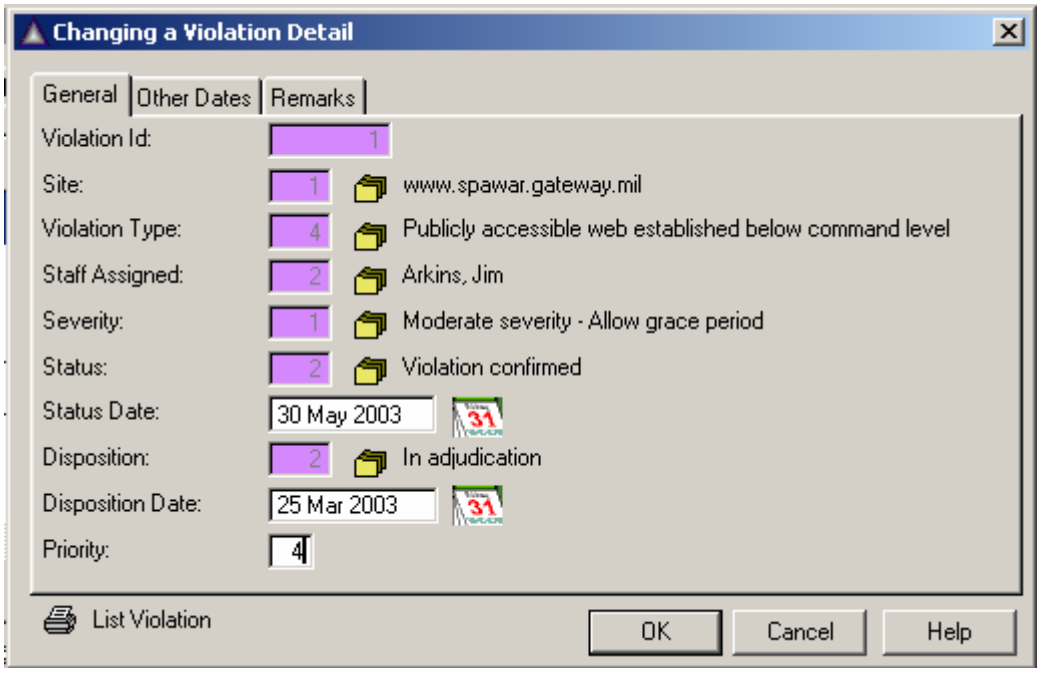


Figure 4 - Violation Update

The update form provides a report button which, when clicked, initiates printing of all information pertinent to the selected violation only. Note the use of drop lists allowing the inclusion of complex information at a minimal cost in user input time and data storage space.

4. Execution Speed and Efficiciency – Unlike some RAD tools, a Clarion application compiles into a true executable (.exe or .dll), maximizing the execution speed of the application.

5. Enhancement Flexibility and Adaptability – It is axiomatic that no organization stands still. This is acutely true whenever information sciences are concerned. Constantly changing commands, missions, weaponry, and assignments – both defensive and offensive – dictate that flexibility and ease of modification be essential features of any application constructed for Naval information support.

Clarion offers such flexibility. Here are some ways the FIWC Website Compliance Application can be modified to accommodate changing or redefined needs:

a. Database structure can be changed any time just by modifying the schema from the Clarion IDE. Such modifications include, but are not limited to: re-characterizing attributes (e.g., from integer to string, short integer to long,

changing length of a string from 60 to 120 etc.); adding and deleting tables; adding and deleting attributes; adding and deleting table relational bindings.

b. After structural changes to the schema, when brought up in the IDE, the application can be "synchronized" – at the application, form, or control level – to incorporate the database modifications automatically.

c. Browse, update, select and report forms are easily added to the application, either by wizard with subsequent manual modification or entirely by hand.

d. The client-server based application can be web-enabled for Internet or intranet access with minimal effort.

e. Controls such as entry fields, drop lists and buttons can be given intelligent behavior by embedding control-linked code at any of several, event-triggered points.

f. A form's "look and feel" is easily altered to provide rich functionality to users. Such modifications include adding a tree browse for Commands; adding tabs to a browse; changing an attribute's update paradigm; and user-interfaces that add range and/or other selection criteria for a report.

B. THE APPLICATION

1. Functional Description

There are sixty-four FIWC Website Compliance application procedures. These sixty-four procedures break down into categories as shown in Table 2 - Distribution of Procedures:

Type	Number
Splash Screens	1
Login Windows	1
Main MDI[32] Forms	1
Browse Forms	14
Update Forms	12
Select Windows	10
Report Procedures	25
Total Procedures:	64

Table 2 - Distribution of Procedures

Browse and update forms correspond directly to the twelve tables in the database. Two additional prototype browse forms support to issuance of memoranda of violation and directed messages (please refer to Application-Level Unimplemented Features later in this chapter). Select windows also correspond to tables, but two tables (Chains and Violations) are primary and therefore never selected. Of the sixty-four procedures, only a few merit special attention here, although all are identified in Appendix G. The following paragraphs address the following aspects of each procedure:

- Its function;
- Application forms and procedures with which it has a direct relationship;
- Special logic and code embedded within the window and / or controls;

[32] MDI stands for Multiple Document Interface. There is usually exactly one MDI form in a client-server model application.

- Unimplemented features which are deemed either essential or enhancing to a full, production application.

2. Login Window

Figure 5 - Login Dialog

Main Function: To identify and authenticate application users.

Procedural Notes: After the user has entered both the registered user identification number (*Staff Id*) and an up-to 20-character password, and has either clicked "Login" or hit the Enter key, the procedure looks up the entered Id / Password pair in the encrypted *Chains* table. Either unregistered user identification or invalid password results in the dialog refusing entry and offering another chance to log in.

The privilege class (presently, either "Normal" or "Supervisory") registered for each unique user determines whether certain designated system functions are executable. At present, only Password Browse and Password Update procedures are so restricted (i.e., accessible only by supervisory users).

Directly Related Procedures: The login dialog is invoked as a thread from, and is subordinate to, the main MDI window. This is its only relationship and the only way it can be executed.

Special Embedded Code & Logic: A loop construct, which validates both the user identification and the associated password against the *Chains* table.

Unimplemented Features: A limit on the number of times a user can enter an invalid user identification and / or an invalid password (the usual limit is three). The purpose of a limit would be to impede attempts to brute-force the login.

3. Main MDI Form

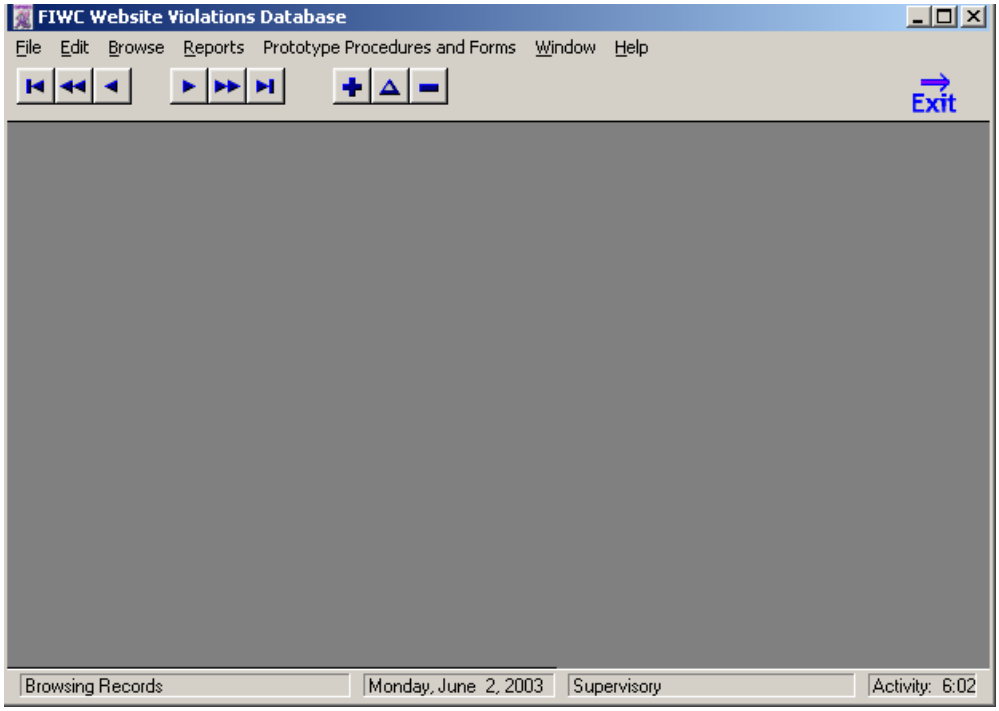


Figure 6 - MDI Window[33]

Main Function: The MDI Window has six major functions:

1. Provides context framework for all visible procedures in the application (including the login window).
2. Restricts secured procedure access to supervisory users;
3. Provides a menu from which users can invoke various application procedures;
4. In addition to the controls on each browse form, gives the user an alternative set of browse navigation and insert, modify, and delete buttons.
5. Provides an exit button to terminate the application.
6. Tells active thread function, today's date, executing privilege, and time current activity in execution on the status bar (bottom of form).

Procedural Notes: None.

[33] Note: The MDI Window is maximized, so in practice it consumes the entire display.

Directly Related Procedures: The login window and all browse and report procedures are invoked as threads from the MDI form. When a thread is terminated, control returns to the MDI window. A FIWC splash panel is invoked from this form at the beginning of each session immediately before the login dialog is processed.

Special Embedded Code & Logic: Logic to display current date and time in the "app-frame" (bottom) section of the window. Logic to invoke the FIWC splash panel. Logic to invoke the login thread. Logic to restrict access of secured procedures to supervisory users.

Unimplemented Features: None.

4. Violations Browse Form

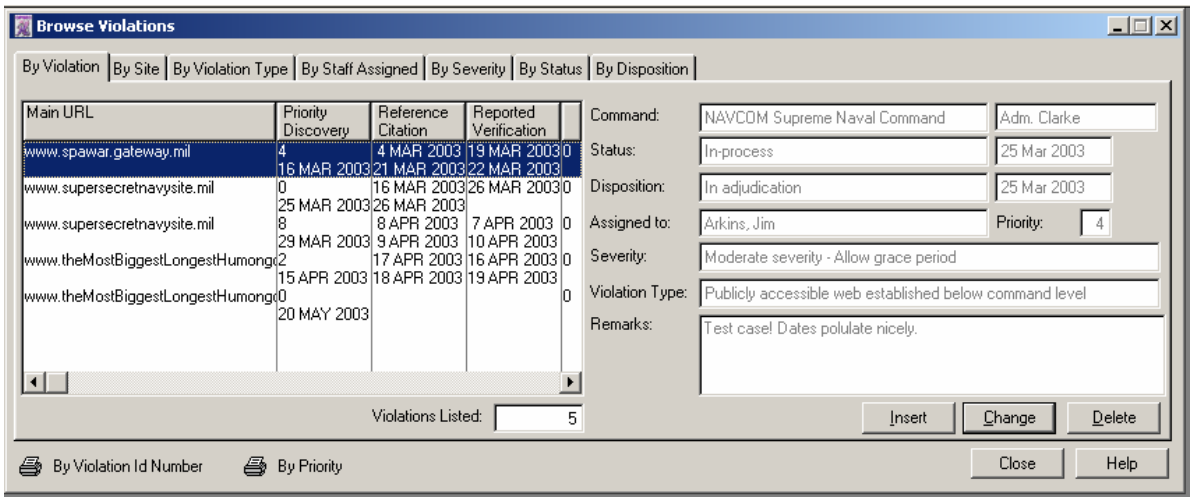


Figure 7 - Violations Browse

Main Function: To provide a means of viewing the Violations table.

Procedural Notes: This layout is typical of the more involved browse forms. The list box on the left side shows selected endemic[34] table attributes. Fields on the right-hand side of the window for the most part show virtual content, i.e., information obtained from linked reference tables. For example, Disposition in the Violations table is an integer, which is rather meaningless to the user. The Disposition field on the right, however,

[34] An "endemic attribute" is one whose content lies entirely within the table of residence. For example, *Main URL* and *Discovery Date* appear in the list box exactly as they are in the table. This contrasts with "virtual attributes" – typically descriptions – derived from reference tables. The endemic attribute in such cases is a foreign key that links to the reference table's primary key. From the users' perspective, the distinction between endemic and virtual attributes is moot, as it should be.

displays the description obtained from the appropriate Dispositions tuple. Information shown on the right is synchronized automatically to the list-box row with focus.

Directly Related Procedures: The *Update Violations* procedure / form is invoked when the user clicks the Insert, Change, or Delete button. Two report procedures are launched from the *By Violations* tab, as depicted. Each of the other six tabs has a single report button that invokes a report ordered according to the tab on which it appears. These six minor tabs also provide select buttons. When clicked, the select button brings up a form from which the user may choose the category of interest. For example, on the *By Severity* tab, a *Select Severity* button lets the user refine the list-box view to violations with the selected severity.

Special Embedded Code & Logic: Refresh of virtual fields' content. Interface to subordinate report, update, and select procedures.

Unimplemented Features: None.

5. Browse Parent Commands Form

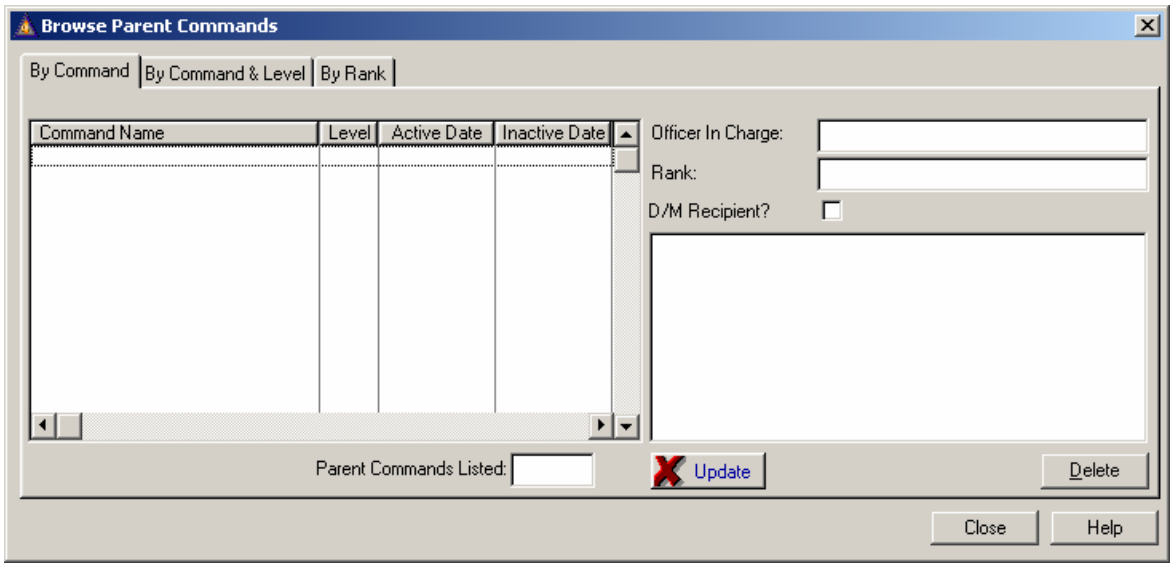


Figure 8 - Parent Commands Browse

Main Function: To view or delete tuples from the Parent Commands table.

Procedural Notes: Although they are two distinct tables, Commands and Parent Commands are procedurally bound. All insertions and modifications to a given Commands tuple, by means of embed coding, similarly affect the corresponding Parent

Command tuple, thus keeping the two tables in synchronization. The main point of distinction is that Parent Commands is a subset of Commands because "leaf node" commands which lack subordinate commands would not be present in the Parent Commands table. Implementation of this subset constraint is discretionary. In other words, hiding and disabling the Parent Commands delete button would keep the two tables in absolute parity.

Directly Related Procedures: Update Parent Commands for deletion only. Note that Change and Insert functions are disabled in this form.

Special Embedded Code & Logic: None

Unimplemented Features: The Delete button may be silenced (i.e., disabled and hidden) to prevent unintentional deletion of a Parent Command. Ideally, there should be a tab showing an indented tree structure for the selected Parent Command's subordinate commands. The tree-format browse view is supported in Clarion. Please refer to Application-Level Unimplemented Features and Appendix J for discussion on implementation of an indented browse tree.

6. Violations Update Form

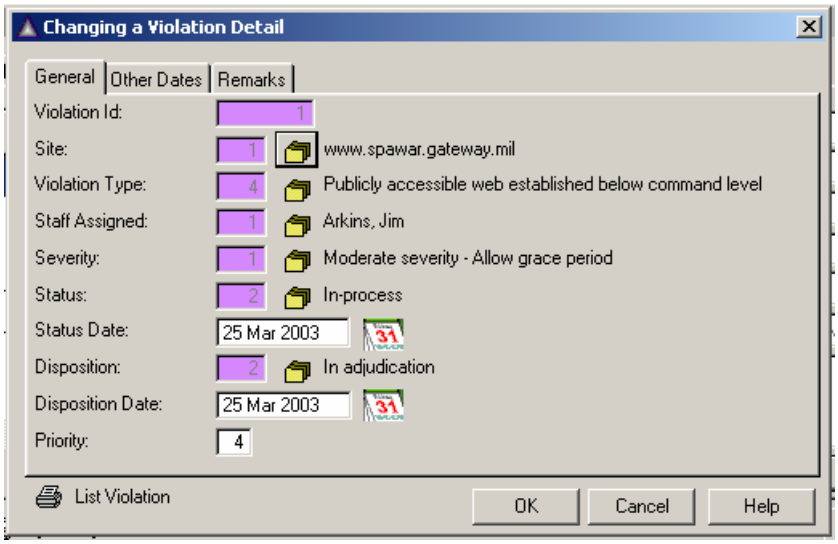


Figure 9 - Violations Update Form

Main Function: To support insertions and changes to Violations tuples.

Procedural Notes: Attributes are distributed among three tabs. Those of a general nature appear on the first or "General" tab. There is only one manual entry field on this tab: Priority. But even Priority can be copied from the corresponding *Violation Type* tuple based on a dialog prompt when *Violation Type* is selected. The rest of the fields are updated: automatically (*Violation Id*), by reference selection (e.g., Site, Violation Type, Staff Assigned, etc.), by calendar selection (e.g., Status Date, Disposition Date), and by default (Status Date, Disposition Date, and Discovery Date), although all dates can be entered manually as well. The second tab contains Discovery Date, Reported Date, Reference Date, Citation Date, and Verification Date. Like the dates depicted in Figure 9 - Violations Update Form, these dates are similarly updatable either by manual entry or by calendar selection.

Finally, the violation of focus can be printed by clicking the List button in the lower left corner of the form.

Directly Related Procedures: Print Violation by Violation Id (subordinate); Browse Violations (Superior).

Special Embedded Code & Logic: All logic to invoke select forms and to expand and display selected reference descriptions on the form. Logic to reject irrational dates, e.g., a Citation Date that occurs before the Discovery Date.

Unimplemented Features: None.

B. APPLICATION-LEVEL UNIMPLEMENTED FEATURES

The following paragraphs discuss features which, although essential or beneficial in a production application, are not fully realized in the prototype implementation.

1. Automatic Memorandum of Violation and Grace Period: Current FIWC policy dictates informal e-mail contact by FIWC to the violating site's webmaster. The email advises the webmaster of violation particulars and provides a copy of an assessment report. A grace period of 30 days is granted for correcting the cited violations.

2. There are three automated steps in this notification process: First, candidate sites must be identified; second, memoranda of violation must be created; and finally, the memoranda of violation must be dispatched to

the offending sites' webmasters. Ideally, all three steps can and should be integrated into the FIWC Website Compliance Application.

Identification: A dual-list box browse form must be developed to highlight *Memorandum of Violation and Grace Period* candidate sites for which notices have not yet been generated or queued.[35] Candidate sites are those with violations having a unique Status[36], e.g., "Violation confirmed." The browse will present violations by site, ordered by Status-Date. In this case, Status Date is the date on which the status "Violation confirmed" was selected from the Statuses reference tuple. The following figure depicts this browse[37]:

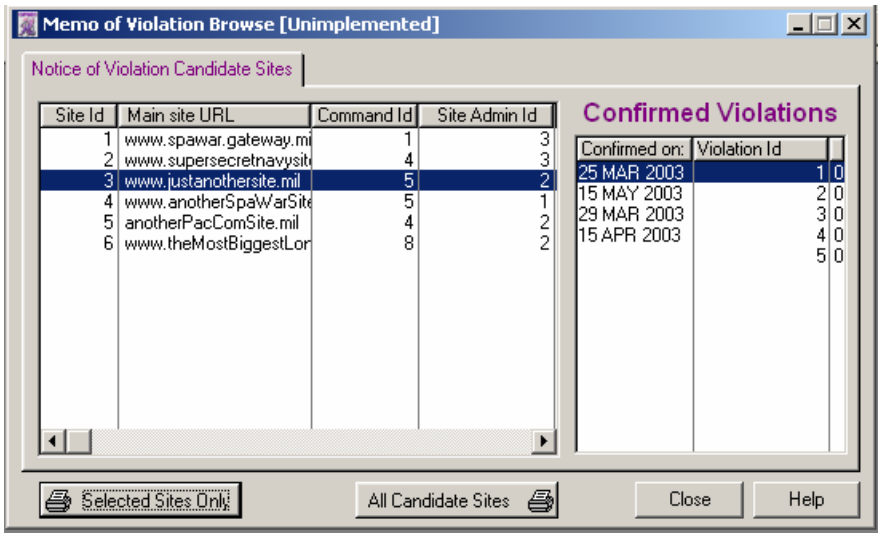


Figure 10 - Memo of Violation Browse

The "Confirmed Violations" list box is linked to show all confirmed violations for whichever Candidate Site is highlighted in the list box on the left.

[35] This form is partially implemented. Primary (Sites) and secondary (Violations) list boxes are synchronized.

[36] *Status* is a foreign-key attribute in the *Violations* (i.e., *VioDetail*) table, selected from the *Statuses* table.

[37] The form has been implemented as depicted in the prototype application. However none of the logic essential to proper execution has been coded.

Creation: Clicking either the "Selected Sites Only" button or the "All Candidate Sites" button would invoke a special procedure which would create properly addressed and dated, hard- and/or electronic-copy memoranda of violation. Upon completed execution of the creation thread, processed violations' statuses should be changed to a valid new Status (e.g., "Notified of Violation"). This change should be an automatic function, performed either by the creation or by the dispatch procedure, depending upon implementation details yet to be determined.

Dispatch: Printing and/or emailing memoranda of violation could be integrated into the creation step. If preferred – for review or other purposes – transient candidate tuples could be queued in a special table (e.g., "*MemorandaPendingDispatch*") for deferred update of Status and dispatch. These transient tuples would be removed once the update and dispatch procedures were completed.

1. Automatic Directed Message Issuance:

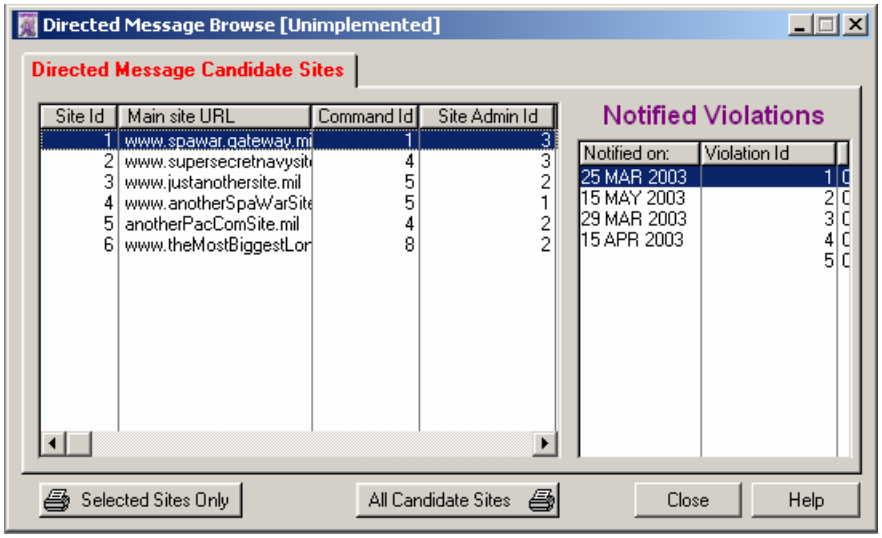


Figure 11 - Directed Message Browse

A thirty-day grace period begins once a *Memorandum of Violation and Grace Period* has been dispatched to the offending site's webmaster. Upon expiration of this thirty-day grace period, current FIWC policy

requires that a directed message be dispatched to the ranking, one- or two-star admiralty with command responsibility over the offending site.

A second not fully implemented browse form[38] would provide a mechanism for the issuance of these directed messages. Sites with violations whose "Notified" status was stamped thirty or more days ago would appear automatically on the Directed Message Browse's main list-box.

In view of the gravity of a directed message, certain automated safeguards should be programmed into the application to prevent false positive[39] directed messages. One such automated safeguard would be to designate the Directed Message Browse as restricted. A restricted procedure requires that the user possess supervisory password access to invoke it. Another automated safeguard would be simply confirming intent with a "yes / no" dialog when either of the dispatch buttons is clicked (i.e., before generating directed messages). Of course, other precautions such as supplementary review by FIWC command staff may also be imposed.

Procedures for identifying, creating, and dispatching directed messages are similar to those for memorandum of violation, except that the creation step must search the command hierarchy, beginning with the command in violation, until it finds a three- or four-star command (distinguished by a value of true in the binary attribute *Com:DirectedMessageRecipient*). Directed message creation would feed TurboPrep[40].

2. Defaults and Drop Lists: One complaint with respect to the existing system is that "five discrepancies found on one URL require five distinct entries." On one hand, it is important to retain full detail on distinct violations, even if several are discovered at the same time on the same

[38] This form is partially implemented. Primary (Sites) and secondary (Violations) list boxes are synchronized.

[39] A "false positive" in this context is the mistaken issuance of a directed message to a command with in fact no sites in violation.

[40] TurboPrep is the message formatting software used for all DMS transmittals.

website. On the other hand, the repeated entry of multiple incidents having common information can be time consuming and can produce inconsistencies. Similarly, multiple notifications of violation and directed messages for a single site can be unduly labor intensive. We believe that this or a similar implementation accommodates these concerns by reducing superfluous labor to an absolute minimum.

This labor reduction is accomplished on the data entry side by drop lists, calendar buttons, automatic fields, and defaults when creating violations tuples. The process can be further streamlined by the simple expedient of tagging the insert procedure as iterative (i.e., the form doesn't automatically shut down after the insertion is complete, but rather refreshes the form and awaits either a new insertion or clicking of the cancel button). In conjunction with making inserts iterative, certain data attributes should be designated persistent on the insert instance of the update form. This would enable the user to simply tab past fields already containing the desired value carried forward from the previously entered violation. Implementation of these two enhancements should minimize substantially both the incidence of data entry inconsistencies and the time and effort required to add violations to the database.

The issuance of memoranda of violation and directed messages should also benefit, both in accuracy and labor intensiveness, from the suggested automation.

3. Database Segmentation: This enhancement would segment violations by assigned staff person such that non-supervisory users would see and have access to only those violations to which they had been assigned. To the non-supervisory user, a segmented database appears exclusively dedicated. Users with supervisory privilege passwords, however, would still have full access to all violations and procedures. The existing password dialog already captures the user's Staff Identification number and stores it in the session-global item *Glo:StaffId*.

The following relatively simple steps are required to implement database segmentation:

a. Violation tuples contain a foreign key (*Vid:StaffId*). Imposing segmentation on the Violations Browse is simply a matter of coding a filter (*Vid:StaffId* <> *Glo:StaffId* and *Glo:PrivilegeLevel* <> 'Supervisory') on the browse procedure.

b. The Browse Staff procedure must be restricted in the main MDI form using the same logic by which Browse Passwords is presently restricted.

c. Finally, add a restriction just like the one implemented for Browse Staff in the main MDI procedure to similarly restrict access to the Select Staff form.

d. The Update Violations procedure must be modified for restricted users. If the user is restricted (i.e., non-supervisory), instead of using the drop list for assigning *Vid:StaffId* from the Staff table, the *Vid:StaffId* is updated automatically on insert with the simple statement *Vid:StaffId* = *Glo:StaffId*. In this case, the drop-list button control is hidden and disabled.

4. Relational Integrity: The database accommodates a hierarchical command structure through the defined relationship between *Commands* and *Parent Commands* tables. This functionality should be enriched by implementing the following features:

   a. To give the application proper relational integrity, prevent selection of a parent command whose level is less than[41] that of the command of focus. This constraint would be implemented in the Update Commands procedure *Parent Command* button ("control event handling accepted" event).

[41] Implementing the absolute ascendant (i.e., no lateral-level linkages) would eliminate the possibility of erroneous recursive relationships. The ascendancy rule would have to apply, not only when selecting a parent command, but also when changing the command level.

b. Implement the "Tree" tab in the Browse Parent Commands procedure. Instead of a list box control, which appears on most application browse tabs, this tab presents an indented hierarchy of commands subordinate to the parent command with focus. Please refer to Appendix J for background correspondence between the author and certain Clarion developers on implementation techniques.

c. Implement a report that presents violations in a hierarchical manner, with violation counts for every command control break.

5. Alarms: Add popup and, optionally, audible alarms for user-specified events and followup actions. An alarm would conditionally highlight a command or violation with an action cue (e.g., "Find out whether new webmaster has been assigned and, if so, journal it."). For popup alarms to work, the application must be active (i.e., on the task bar, if quiescent.

6. Report Filters: At present, the violation reports by browse tab (e.g., by Status, by Disposition, etc.) are grouped by the category shown on the tab. In practice, they should be filtered using the same criterion as the one in the tab from which the report is being launched. This is not a particularly difficult feature to implement, requiring only passing of the filter and its key name, and the creation of a single, generalized violations report (through cloning) which implements the tab categories.

7. Additional Encryption: The password table (Chains) is already encrypted[42]. If security concerns dictate, any (or all) of the other tables may be encrypted as well. Although this encryption does not meet rigorous formal security specifications, it does offer protection above that provided by a plaintext table. The tradeoff is processing efficiency.

   The password table is small and infrequently referenced, so encryption offers obvious advantages at negligible cost. On the other hand, encrypting a large-volume and volatile table such as Violations involves more

[42] This is an undocumented and unidentified, proprietary encryption algorithm offered as part of the Clarion development platform for TopSpeed tables only.

ponderous processing considerations, which should be weighed against the risks and consequences of a security breech.

A third-party developer, Brady and Associates, LLC[43], provides common MD5 encryption for the Clarion IDE. This encryption is implemented in dlls. In addition, it is rumored that another third-party developer is implementing Blowfish encryption for Clarion, but no details of this implementation are known to the author.

8. Drop-list Filters: All reference tables contain the attributes *Active Date* and *Inactive Date*. Their purposes are: 1) to leave an audit trail of the date on which a reference tuple became active[44] and, if no longer active, when it became inactive, and 2) to provide a means of deactivating the reference tuple for drop lists without orphaning it in reports and views.

   For example, say a particular FIWC staff person is reassigned. Although no longer selectable for new violations, s/he should continue to exist in the Staff table so that previously assigned violations will show the staff person in reports and views. This enhancement is quite easy to implement.

[43] http://www.clariondeveloper.com

[44] *Active Date*, in all cases, defaults to the date on which the tuple was created.

THIS PAGE INTENTIONALLY LEFT BLANK

V – CONCLUSION

This thesis focused on FIWC's six main areas of website oversight and enforcement activity:

3. Assessment and Discovery
4. Documentation
5. Review
6. Record-keeping
7. Citation
8. Verification

Of these, the proposed architecture and solution address all but the first. Assessment and Discovery was judged outside the scope of the thesis, being exclusively concerned with roaming the web in search of publicly accessible Navel websites and reviewing these sites for content fitting pre-defined violation profiles.

The remaining five – Documentation, Review, Record-keeping, Citation, and Verification – were deemed appropriate to our thesis objectives. In the final analysis, we believe that the research, findings, and architecture embodied in this thesis should contribute to the implementation of a production-class solution for FIWC.

A. EVOLVING PERCEPTIONS

At the outset, we had only a general idea of the problem we were attacking, and little notion of the form our final implementation would take. But, as the research progressed, our problems and challenges became more distinct and the solution began to take form.

Our initial working concept of the database consisted of fifteen unclassified attributes. From this beginning, a database with 12 tables, an aggregate of 95 attributes, and 30 inter-table relationships evolved. In the final prototype implementation, there are

64 procedural modules, each performing a unique application task supporting table creation, maintenance, viewing and reporting.

Our first challenge was to design a database whose organization, content, and infrastructure would support the information demands of the application. As outlined in Chapter III – Architecture Design, the construction of the database was a twelve-step process. We knew that, in a datacentric application such as this, thoughtful and thorough design of the database is the cornerstone of a sound implementation.

Once this initial database design was in place, procedural development was begun. Thanks to the Clarion RAD tool, we were able automatically to generate skeletal procedures from our schema and dictionary. The amount of detail effort this displaced cannot be overemphasized.

The 64 skeletal procedures, each supporting a window populated with schema-derived controls, formed the procedural framework for our application.

The next task was to develop the skeletal forms and procedures into functioning units. This involved tasks which ranged from the routine (e.g., placement and sizing of controls, wording and titles, list box margins and content) to more challenging activities (e.g., imposition of a password portal, supervisory restrictions, synchronization of subordinate browse views, initializing virtual content for updates and browses, and the creation of active drop lists for all update forms).

We were fortunate in having only one false start during the entire project. Our original approach to the command tree was to use a single *Commands* table, each tuple containing two keys: *Command Id* and *Parent Command Id*. Although this was an elegant and economical concept, we realized early on, that implementing tree relationships with this single-table construct would exceed the allowed thesis timeframe. We reluctantly abandoned this approach for a more easily implemented, albeit less elegant, two-table solution. using *Commands* and *Parent Commands*. To shelter the user from the burden and hazards of coordinating maintenance of these two tables, we internalized synchronization, automatically updating *Parent Commands* whenever there is a modification or insertion to *Commands* and by disabling direct (i.e., user explicit) update of *Parent Commands*.

The last procedural enhancement was the addition of the password facility. This required a simple change to the schema, some dictionary modifications, the creation of three procedures, and writing code for the main window to deploy the stand-alone login procedure.

B. WHAT REMAINS

As mentioned in Chapter IV – Implementation, there are certain enhancements which should be considered for implementation in order to bring the prototype into production-class status. The following is a recap of these enhancements, including the author's rough estimate of the person hours required to implement and test them. The first two enhancements listed in Table 3 - Application Enhancements are specific to the existing procedures, Login and Parent Commands Browse. The rest are Application Level enhancements (i.e., they involve new procedures and/or span several existing procedures and possibly the schema).

Enhancement	Person Hours
Login attempt limitation	12
Indented tree graph display on Parent Commands browse	50
Automatic Memo of Violation	40
Automatic Directed Messages	70
Iterative *Violations* inserts	2
Persistent *Violations* data-entry fields	4
Database Segmentation	25
Mods to prevent incoherent / recursive command chains	30
Command-tree list box in Browse Parent Commands	90
Popup alarms for followup events	60
Browse tab filters on Violation reports	40
Encryption of certain database tables	8
Select form filtering on *Inactive Date*	20

Table 3 - Application Enhancements

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDICES

The following pages contain documentation which, while not essential to the main body of the thesis, serves as reference material for positions, assertions, and attributions contained therein.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A
DOD WEB SITE ADMINISTRATION MESSAGE

Web Site Administration

Policies & Procedures

November 25, 1998

With Amendments and Corrections incorporated in red italics

(latest corrections from 11 January, 2002)

Office of the Assistant Secretary of Defense

(Command, Control, Communications & Intelligence)

6000 Defense Pentagon

Washington, DC 20301-6000

Department of Defense

WEB SITE ADMINISTRATION GUIDANCE

CONTENTS

DEPSECDEF Memorandum Subject: Web Site Information Services DoD-Wide, dated November 25, 1998 implements the policies, responsibilities and procedures for Web Site Administration. An electronic copy of this guidance is available at http://www.defenselink.mil/admin/about.html#WebPolicies. Please forward comments, suggestions and recommendations for changes to: OASD (C3I), ODASD (Policy & Implementation/Deputy CIO), 6000 Defense Pentagon, Washington, DC 20301-6000.

WEB SITE ADMINISTRATION

Part I - Policy & Responsibilities

November 25, 1998

1. PURPOSE

This document delineates the policy and assigns responsibility related to establishing, operating and maintaining unclassified Web sites and other related services. It supersedes the "Guidelines for Establishing and Maintaining a Publicly Accessible Department of Defense Web Information Service" jointly published by the Office of the Assistant Secretary of Defense (Public Affairs) and the Office of the Assistant Secretary

of Defense (Command, Control, Communications and Intelligence) on July 18, 1997 (updated January 9, 1998).

2. APPLICABILITY

This policy applies to:

2.1. The Office of the Secretary of Defense (OSD), the Military Departments (including the Coast Guard when it is operated as a Military Service in the Navy), the Chairman of the Joint Chiefs of Staff, the Combatant Commands, the Defense Agencies, and the Department of Defense (DoD) Field Activities (hereafter referred to collectively as "the DoD Components") and to their contractors and consultants including those who operate or maintain DoD Web sites for them, through incorporation into contracts.

2.2. All unclassified DoD Web sites, both publicly and non-publicly accessible.

2.3. Reviewing approval requests received from DoD contractors and subcontractors relative to the posting of unclassified DoD information to a DoD contractor Web site.

3. DEFINITIONS

Terms used in this document are defined in Part III.

4. POLICY

It is the policy of the DoD that:

4.1. Using the World Wide Web is strongly encouraged in that it provides the DoD with a powerful tool to convey information quickly and efficiently on a broad range of topics relating to its activities, objectives, policies and programs.

4.2. The considerable mission benefits gained by using the Web must be carefully balanced through the application of comprehensive risk management procedures against the potential risk to DoD interests, such as national security, the conduct of federal programs, the safety and security of personnel or assets, or individual privacy created by having electronically aggregated DoD information more readily accessible to a worldwide audience.

4.3. Each organization operating a DoD Web site will implement technical security best practices with regard to its establishment, maintenance and administration.

4.3.1. DoD Web sites containing i) FOR OFFICIAL USE ONLY information, ii) information not specifically cleared and marked as approved for public release in accordance with DoD Directive 5230.9 and DoD Instruction 5230.29 (references (h) and (o)), or iii) information of questionable value to the general public and for which worldwide dissemination poses an unacceptable risk to the DoD, especially in electronically aggregated form, must employ additional security and access controls. Web sites containing information in these categories should not be accessible to the general public.

4.4. Consistent with other leadership responsibilities for public and internal communication, the decision whether or not to establish an organizational Web site, and to publish appropriate instructions and regulations for a Web site within the limitations established by this document, is hereby delegated to each DoD Component.

5. RESPONSIBILITIES

5.1. The Assistant Secretary of Defense for Command, Control, Communications and Intelligence (ASD (C3I)) shall:

5.1.1. Provide policy and procedural guidance with respect to establishing, operating and maintaining Web sites.

5.1.2. Maintain liaison with the Assistant Secretary of Defense for Public Affairs to provide policy oversight and guidance to ensure the effective dissemination of defense information via the Internet.

5.1.3. Provide technical support consistent with existing Chief Information Officer (CIO) responsibilities.

5.1.4. Develop and maintain, in coordination with the Chairman of the Joint Chiefs of Staff, Under Secretary of Defense (Personnel &Readiness), and General Counsel, training guidance and requirements that addresses information security on the Web.

5.1.5. Approve and publish DoD Instructions and Publications, as necessary, to guide, direct, or help Web site activities, consistent with DoD 5025.1-M (reference (kk)).

5.1.6. Provide a mechanism for feedback reporting across DoD, to include "Lessons Learned" and the identification of useful automated tools to aid in the conduct of multi-disciplinary security assessments of Web sites.

5.1.7. Ensure compliance with this policy.

5.2. The Assistant Secretary of Defense for Public Affairs (OASD (PA)) shall:

5.2.1. Operate and maintain DefenseLINK (http://www.defenselink.mil) as the official primary point of access to DoD information on the Internet.

5.2.2. In coordination with the other OSD Principal Staff Assistants, provide oversight policy and guidance to ensure the absolute credibility of defense information released to the public through publicly accessible Web sites.

5.2.3. Establish and maintain a central Web site registration system for the Department that meets the requirements for the Government Information Locator Service (GILS) and is integrated with Service-level registration systems.

5.3. The Assistant Secretary of Defense for Reserve Affairs and the Chairman of the Joint Chiefs of Staff shall develop and implement a plan that uses Reserve Component

assets to conduct ongoing operations security and threat assessments of Component Web sites.

5.4. The Secretaries of the Military Departments shall establish and maintain a central registration system for the respective service that meets the requirements for GILS and is integrated with DefenseLINK.

5.5. The Heads of the DoD Components shall:

5.5.1. Establish a process for the identification of information appropriate for posting to Web sites and ensure it is consistently applied.

5.5.2. Ensure all information placed on publicly accessible Web sites is properly reviewed for security, levels of sensitivity and other concerns before it is released. Detailed requirements for clearance of information for public release are located in DoD Directive 5230.9 and DOD Instruction 5230.29 (references (h) and (o)) and Part II of this document.

5.5.3. Ensure approved DoD security and privacy notices and applicable disclaimers are used on all Web sites under their purview.

5.5.4. Ensure all information placed on publicly accessible Web sites is appropriate for worldwide dissemination and does not place national security, DoD personnel and assets, mission effectiveness, or the privacy of individuals at an unacceptable level of risk.

5.5.5. Ensure procedures are established for management oversight and regular functional review of the Web site.

5.5.6. Ensure operational integrity and security of the computer and network supporting the Web site is maintained.

5.5.7. Ensure that reasonable efforts are made to verify the accuracy, consistency, appropriateness, and timeliness of all information placed on the Web site.

5.5.8. Register each publicly accessible Web site with the Government Information Locator Service (GILS).

5.5.9. Provide the necessary resources to adequately support Web site operations to include funding, equipping, staffing and training.

5.5.10. Ensure that a comprehensive, multi-disciplinary security assessment is conducted of their Web sites within 120 days of the promulgation of this document, and at least annually thereafter.

5.5.11. Provide a mechanism for feedback reporting within the Component, to include "Lessons Learned" suitable for all DoD Components.

5.5.12. Ensure compliance with this policy for those functions, missions, agencies, and activities in their purview.

5.5.13. Grant waivers on a non-delegable basis to a provision of the procedures contained in Part II of this document when it has been determined that immediate implementation would adversely impact essential mission accomplishment. Instances where such waivers have been granted will be reported to the Assistant Secretary of Defense (C3I).

6. EFFECTIVE DATE. This policy is effective immediately.

Author's Note: Part II through Part V deleted.

Appendix B
FIWC Administrative Message

ADMINISTRATIVE MESSAGE[45]
ROUTINE
R 121301Z JUN 01 ZYB PSN 605425E22
FM FLTINFOWARCEN NORFOLK VA//N3//

TO ALCND

INFO CNO WASHINGTON DC//N3/N5/N6/N64// CNO WASHINGTON DC//N3/N5/N6/N64// SECNAV WASHINGTON DC//JJJ// USCINCSPACE PETERSON AFB CO//J5/J6/J34/J39// CINCLANTFLT NORFOLK VA//N1/N3/N6/N39// CINCPACFLT PEARL HARBOR//N3DC/N6/N69// COMUSNAVCENT//JJJ// COMUSNAVCENT//JJJ// CNET PENSACOLA FL//JJJ// CNET PENSACOLA FL//JJJ// JTF-CND WASHINGTON DC//J3/J6// COMNAVSECGRU FT GEORGE G MEADE MD//N3/N5// COMNAVRESSECGRUCOM FT WORTH TX//JJJ// DIRNAVCRIMINVSERV WASHINGTON DC//20// CHINFO WASHINGTON DC//JJJ// CHINFO WASHINGTON DC//JJJ// CTF-NMCI WASHINGTON DC//00/01/N2/N3// NCTF-CND WASHINGTON DC//N3/N5// USMC NOC QUANTICO VA//JJJ// USMC NOC QUANTICO VA//JJJ// AFIWC KELLY AFB TX//EAA// ACERT FT BELVOIR VA//JJJ// DISA WASHINGTON DC//ASSIST//

THIS IS A 2 SECTIONED MSG COLLATED BY MDS UNCLAS//N05510// ALCND 042/01

MSGID/GENADMIN/FLTINFOWARCEN//

SUBJ/FLTINFOWARCEN (FIWC) WEB RISK ASSESSMENT CRITERIA//

REF/A/RMG/CNO/261622ZMAR99//

REF/B/DOC/SECNAV/SECNAVINST 5720.47/01JUL99//

REF/C/DOC/DEPSECDEF/DEPSECDEF MEMORANDUM/25NOV98//

REF/D/DOC/DEPSECDEF/DEPSECDEF MEMORANDUM/26APR01//

NARR/REF A IS NAVY WORLD WIDE WEB PAGE MONITORING INSTRUCTION.

REF B IS DEPARTMENT OF THE NAVY POLICY FOR CONTENT OF PUBLICLY ACCESSIBLE WORLD WIDE WEBSITES

REF C IS DEPARTMENT OF DEFENSE WEB SITE ADMINISTRATION POLICIES AND PROCEDURES.

REF D PROVIDED ADDITIONS TO REF C.// PPOC/PAUL/LT/FLTINFOWARCEN/-/TEL: (757)417-4179 EXT3/ EMAIL/RPAUL@FIWC.NAVY.MIL//

[45] http://www.seabee.navy.mil/help/Instruct/WEB_RISK_ASSESSMENT.htm

SPOC/NALLEY/CTM1/FLTINFOWARCEN/-/TEL: (757)417-4179 EXT3/
EMAIL/SNALLEY@FIWC.NAVY.MIL//

RMKS/

1. THIS IS A NAVCIRT/NCTF COORDINATED MESSAGE.

2. IAW REF A, FIWC IS RESPONSIBLE FOR CONDUCTING RANDOM WEB SITE VERIFICATION CHECKS AND PROVIDING NON COMPLIANT COMMANDS WITH SPECIFIC DATA CONCERNING NON COMPLIANCE. PREVIOUSLY, THE AREA OF OPERATIONAL SECURITY (OPSEC) WAS FOCUSED ON ACTIVITIES THAT MIGHT ONLY BE SEEN BY A HUMAN OBSERVER, A SATELLITE, NEWS, ETC. THE NEWEST AREA OF CONCERN AND VULNERABILTY IS THE INTERNET. IN AN EFFORT TO REDUCE THE AMOUNT OF SENSITIVE INFORMATION THAT IS POSTED ON PUBLICALLY ACCESSIBLE WEB PAGES, FIWC WAS TASKED TO ASSESS DON WEB SITES FOR COMPLIANCE WITH APPLICABLE DIRECTIVES.

3. IAW REFS A THROUGH D, FIWC USES THE FOLLOWING CRITERIA IN CONDUCTING WEB RISK ASSESSMENTS:

A. WEB SITES THAT ARE FOUND TO CONTAIN ANY OF THE FOLLOWING WILL RESULT IN AN ASSESSMENT OF NON COMPLIANT.

(1) PLANS OR LESSONS REVEALING SENSITIVE MILITARY OPERATIONS, EXERCISES, OR VULNERABILITIES. EXAMPLE: POSTING PLANS ON HOW A SPECIFIC OPERATION OR EXERCISE WILL BE CONDUCTED.

(2) REFERING TO ANY INFO REVEALING SENSITIVE MOVEMENTS OF ANY MILITARY ASSETS OR LOCATIONS OF UNITS, INSTALLATIONS, OR PERSONNEL WHERE UNCERTAINTY REGARDING LOCATIONS IS AN ELEMENT OF THE SECURITY OF THE MILITARY PLAN OR PROGRAM. EXAMPLE: POSTING A SHIP'S UNDERWAY SCHEDULE.

(3) LISTING PERSONAL (NON MILITARY) TELEPHONE NUMBERS. EXAMPLE: LISTING AN OMBUDSMAN'S HOME PHONE NUMBER.

(4) UNITS THAT ARE SENSITIVE, ROUTINELY DEPLOYED, OR STATIONED IN FORIEGN TERRITORIES THAT DISPLAY ORGANIZATIONAL CHARTS/ROSTER BOARDS LISTING NAMES AND/OR BIOGRAPHICAL DATA OF INDIVIDUALS OTHER THAN THE CO, OIC, XO, CMC, PAO OR CIVILIAN EQUIVALENT. EXAMPLE: POSTING A DEPARTMENTAL ORGANIZATION CHART WITH NAMES ASSIGNED TO A SPECIFIC AREA OR JOB.

(5) LISTING PERSONALIZED E-MAIL ADDRESS (OTHER THAN .MIL ACCOUNTS). EXAMPLE: LISTING A SERVICEMAN'S PERSONAL HOTMAIL OR E-MAIL ACCOUNT AT HIS HOME.

(6) COMMANDING OFFICERS READING ROOM OR LISTS OF ANY CLASSIFIED INFO. EXAMPLE: POSTING ANY INFORMATION THAT IS FOR OFFICIAL USE ONLY (PLAN OF THE WEEK, PLAN OF THE DAY, SHIP'S SCHEDULE ETC.).

(7) LISTING ANY SSN'S OR DATE OF BIRTH. EXAMPLE: LISTING THE COMMANDING OFFICER'S DATE OF BIRTH IN HIS BIO.

(8) LISTING NAMES, LOCATIONS, OR ANY OTHER IDENTIFYING INFO ABOUT ANY FAMILY MEMBERS. EXAMPLE: COMMAND CHAPLAIN POSTING A WELCOME ABOARD MESSAGE STATING "MY WIFE (NAME) AND I LOOK FORWARD TO YOUR ARRIVAL."

(9) DISPLAYING OF PHOTOGRAPHS WITH NAMES (EXCEPT CO, OIC, XO, CMC, PAO OR CIVILIAN EQUIVALENT). EXAMPLE: PICTURES OF FROCKED SERVICEMEMBERS WITH THEIR NAMES LISTED IN CAPTION BELOW.

(10) LISTING OF HOME ADDRESSES. EXAMPLE: POSTING A COMMAND RECALL BILL/SOCIAL ROSTER WITH FAMILY MEMBERS NAMES AND ADDRESS LISTED.

(11) IF THERE ARE ANY LINKS OFF THE PAGE THAT ARE NOT COMPLIANT, THEN THE PARENT PAGE IS NOT COMPLIANT. EXAMPLE: HAVING A LINK TO A SITE THAT HAS AN EXTERNAL LINK THAT CONTAINS COMMERCIAL ADVERTISEMENTS OR SPONSORSHIPS WITHOUT THE APPROPRIATE DISCLAIMER.

(12) A SITE CONTAINING ANY WRITTEN OR DISPLAYED INFO STATING THAT THE WEBSITE IS BEST VIEWED WITH ANY SPECIFIC BROWSER, A SITE THAT SELECTS OR RECOMMENDEDS A FEATURED SITE, POINTS TO ANY SEARCH ENGINES OR RECOMMEND ANY COMMERCIAL SOFTWARE. EXAMPLE: A STATEMENT THAT STATES "THIS SITE IS BEST VIEWED WITH INTERNET EXPLORER".

(13) NO MATERIALS OR SERVICES SHALL BE SOLD VIA COMMAND WEBSITE. EXAMPLE: SELLING COMMAND BALLCAPS OR COFFEE MUGS.

B. WEB SITES MUST CONTAIN THE FOLLOWING. WEB SITES FOUND WITHOUT THE FOLLOWING WILL RESULT IN AN ASSESSMENT OF NON COMPLIANT.

(1) PRIVACY AND SECURITY NOTICE.

(2) SITE REGISTERED WITH THE GOVERNMENT INFORMATION LOCATOR SERVICE (GILS).

(3) WEBMASTER CONTACT INFORMATION MUST BE EITHER VISIBLE ON THE SITE HOME PAGE OR IN THE SOURCE CODE OF THE HOME PAGE.

(4) STATEMENT THAT THE SITE IS APPROVED BY EITHER THE PAO AND/OR CMD INFORMATION ASSURANCE OFFICER, VISIBLE ON THE HOME PAGE OR IN THE SOURCE CODE.

(5) WEB SITES SHALL CONTAIN LINKS TO THE FOLLOWING SITES: WWW.NAVY.MIL, THE PARENT COMMAND OR ISIC, AND THE NAVY RECRUITING SITE, WWW.NAVYJOBS.COM.

(6) NOTICE STATING THAT THE SITE IS AN OFFICIAL U.S. NAVY WEB SITE.

4. THIS LIST IS NOT ALL INCLUSIVE. ALL WEBMASTERS SHOULD FREQUENTLY REVIEW THEIR WEB SITES FOR COMPLIANCE WITH THE PUBLISHED POLICIES AND PROCEDURES AS SET FORTH IN REFS A THROUGH D.

A. REFS CAN BE FOUND USING THE FOLLOWING URL'S:

(1) REF A: http://WWW.BUPERS.NAVY.MIL/NAVADMIN/NAV99/NAV99088.TX [Note: Extension truncated, should be ".txt"]

(2) REF B: WWW.DEFENSELINK.MIL/ADMIN/ABOUT.HTML#WEBPOLICIES UNDER MILITARY SERVICE POLICIES "NAVY"

(3) REF C: WWW.DEFENSELINK.MIL/ADMIN/DOD_WEB_POLICY_12071998_ .HTML

(4) REF D: WWW.DEFENSELINK.MIL/ADMIN/DOD_WEB_POLICY_12071998_ AMENDMENT.HTM

B. FIWC IS STANDING BY TO ASSIST COMMANDS WITH ENSURING ALL WEB SITES ARE IN COMPLIANCE.

5. TO REQUEST A WEB RISK ASSESSMENT ON YOUR SITE OR ASK QUESTIONS CONCERNING COMPLIANCE ISSUES, SEND AN EMAIL TO WEB-ASSESSMENT@FIWC .NAVY.MIL.

6. THIS ALCND IS CANCELLED FOR RECORD PURPOSES ON 12 DEC 01.//

BT NNNN RTD:000-000/COPIES:

Appendix C
Army Regulation 25–1

The following policy replaces "Guidance for Management of Publicly Accessible U.S. Army Websites," dated 30 November 1998.

Army Regulation 25–1
Army Information Management
Dated 31 May 2002
Effective 28 June 2002

Excerpted. See http://www.usapa.army.mil/pdffiles/r25_1.pdf to view the entire regulation.

6-3.

r. Internet (World Wide Web (WWW), Intranets, and Extranets.

Official Army web sites may exist on any of the above forms of "nets." The use of these "net" communications can support execution of Army missions through information sharing and can save resources currently expended on traditional means of communication. Users are encouraged to make it their preferred and routine choice to access, develop and exchange information. Army web sites must be in compliance with the DoD web site administration policy located at http://www.defenselink.mil/webmasters/ or contained within subsequent DoD directives. In addition, the following Army policies apply:

(1) Access to all forms of "nets" is authorized for all personnel as deemed reasonable by respective managers. Access may be implemented without further justification than this regulation.

(2) The *AKO* at <www.us.army.mil> is the primary portal for Army unclassified intranets and the NIPRNET. The *AKO-S* is the primary portal for classified intranets and the SIPRNET.

(a) As of July 2002, Army web-enabled business applications will be linked to the AKO portal. Initial minimum standard to link applications to AKO is a URL link on The Army Portal. The objective standard to link applications to AKO is to use the AKO directory services for authentication as well as a URL link on The Army Portal.

(b) AKO is responsible for generating user IDs and accounts, performing authentication via secure Lightweight Directory Access Protocol (LDAP) directory services, publishing updates to

the technical mechanism used for directory services, and incorporating appropriate security measures.

(3) FORSCOM (Army Signal Command) manages the ".army.mil" web site assignment of sub-domains requested by other Army organizations. FORSCOM promulgates procedures for Army sub-domain managers, to include assignment, formatting and any centralized registration of addresses for servers, gateways, organizations and individual users.

(4) Since the internet is a public forum, Army organizations will ensure that the commander, the PAO, and other appropriate designee(s) (for example, command counsel, force protection, intelligence, etc.) have properly cleared information posted to the WWW. The designated reviewer(s) will conduct routine reviews of web sites on a quarterly basis to ensure that each web site is in compliance with the policies herein and that the content remains relevant and appropriate. The minimum review will include all of the web site Management Control Checklist items at Appendix (B-4) of this regulation. Information contained on publicly accessible web sites is subject to the policies and clearance procedures prescribed in AR 360-1, Chapter 5, for the release of information to the public. In addition, Army organizations using the WWW will not make the following types of information available on publicly accessible web sites:

(*a*) Classified or restricted distribution information.

(*b*) For Official Use Only (FOUO) information.

(c) Unclassified information that requires special handling, e.g., Encrypt For Transmission Only, Limited Distribution, scientific and technical information protected under the Technology Transfer Laws.

(*d*) Sensitive but unclassified information such as proprietary information, pre-decisional documents, and information that must be protected under legal conditions such as the Privacy Act.

(*e*) FOIA exempt information. This includes a prohibition of lists of names and other personally identifying information of personnel assigned within a particular component, unit, organization or office in the Department of Army. Discretionary release of names and duty information of personnel who, by nature of their positions and duties, frequently interact with the public, such as

flag/general officers and senior executives, public affairs officers, or other personnel designated as official command spokespersons, is permitted.

(*f*) Draft publications. See also paragraph 9-2.

(5) The Army CIO will provide policies, procedures and format conventions for web sites, and will promulgate such guidance in this regulation and on the Army web site http://www.army.mil/webmasters/ .

(6) Army organizations will assign a web master/maintainer for each of their web sites. Army organizations will provide their web masters/maintainers sufficient resources and training. Web masters/maintainers will have technical control over updating the site's content and will ensure the site conforms to Defense and Army-wide policies and conventions.

(7) Organizations maintaining publicly accessible web sites must register the site with the Government Information Locator Service (GILS) at http://sites.defenselink.mil/. GILS is used to identify public information resources throughout the U.S. Federal government.

(8) Organizations maintaining private web sites (e.g., intranets, extranets) must register them with the Army Networks and Systems Operations Center (ANSOC) and assure that the Secure Sockets Layer (SSL) is enabled and that PKI encryption certificates are loaded. PKI web server certificates may be obtained from the Army Network Systems Operations Center (ANSOC), Army Signal Command (ASC) of U.S. Army Forces Command (FORSCOM).

(a) All web applications will support client authentication to the applicable private web server at a minimum.

(b) All unclassified, private Army web servers will be enabled to use DoD PKI certificates for server authentication and client/server authentication. The following type of web server is exempt from this mandate: any unclassified Army web server providing non-sensitive, publicly releasable information resources that is categorized as a private web server only because it limits access to a particular audience only for the purpose of preserving copyright protection of the contained information sources, facilitating its own development, or limiting access to link(s) to limited access site(s) (and not the information resources).

(9) Every Army organization that maintains a public web site must observe Federal, Defense, and Army policies on protecting personal privacy on official Army web sites and establish a process for web masters/maintainers to routinely screen their web sites to ensure compliance. At a minimum, web sites must comply with the following web privacy rules:

(a) Web masters/maintainers will display a Privacy and Security Notice in a prominent location on at least the first page of all major sections of each web site.

(b) Each Privacy and Security Notice must clearly and concisely inform visitors to the site what information the activity collects about individuals, why it is collected, and how it will be used. For an example, see the Defenselink (official web site of the Department of Defense: http://www.defenselink.mil). For management purposes, statistical summary information or other non-user identifying information may be gathered for the purposes of assessing usefulness of information, determining technical design specifications, and identifying system performance or problem areas.

(c) "Persistent" cookies, that is, those cookies that can be used to track users over time and across different web sites to collect personal information, are prohibited. The use of any other automated means to collect personally identifying information without the express permission of the user is prohibited. Requests for exceptions must be forwarded to the Army CIO.

(d) "Third party" cookies will be identified and purged from official web sites.

(10) All Army private (non-publicly accessible) web sites must be located on a ".mil" domain.

(11) Possible risks must be judged and weighed against potential benefits prior to posting any Army information on the WWW. (See also paragraph 5-10).

(12) Web masters/maintainers will provide a "re-direct" page when the URL of the web site is changed.

(13) Army organizations maintaining web sites are required to achieve web site compliance with the provisions of section 508 of the Rehabilitation Act Amendments of 1998. Refer to section 508 standards on Web-based, Intranet, and Internet information and applications at http://www.section508.gov/.

This site offers free on-line training to assist web developers in how to design web sites for 508 compliance. See also paragraph 6-1.i. on information access for the handicapped.

(14) Web sites published by Army commands but hosted on commercial servers (servers other than “army.mil”) are considered official sites and remain subject to this policy.

(15) Army commands and activities will establish objective and supportable criteria or guidelines for the selection and maintenance of links to external web sites. Guidelines should consider the informational needs of personnel and their families, mission-related needs, and public communications and community relations objectives.

p. 66

9-2. Central configuration management.

……..

b. Only those web sites approved by the AASA may host Army-wide departmental publications and forms on their web sites. Those activities desiring to provide internet access to departmental publications and forms on a web site must establish electronic links to the approved official publications and forms web site as listed in the official repository instead of publishing a duplicate publication.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix D
FIWC Registration Form

All Fields Required!!!

***** Command Information *****

Command Full Name: Help

Command PLA: (Plain Language Address) Help

Street Address:

City:

State:

Zip Code:

Country:

Time Zone: Select One

Quarterdeck/24 Hour Telephone Number: Help **Extension**

Command ISIC PLA: Help

Echelon: Help Select One

***** PAO Information *****

PAO Office Code:

PAO Email Address:

PAO Telephone Number: **Extension**

PAO Fax Number:

***** Web Site Information *****

Web Site Home page URL:

Self Assessment complete?: Help

Self-Assessment Checklist (**Word** or **PDF**)

Access Restrictions: Help

Public Access Secured Socket Layer (SSL) Domain Restriction Login Access

Key Words: (List of key words separated by comma's) Help

Webmaster Name:

Webmaster Telephone Number: **Extension**

Webmaster Email:

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix E
FIWC Checklist

WEB SITE SELF ASSESSMENT CHECKLIST

Updated: 7 November 2002

Ref A DoD Web Site Administration Policies and Procedures

Ref B SECNAVINST 5720.47

Ref C NAVADMIN 088/99

Ref D SECDEF Memo 28DEC2001

Ref E SECDEF Memo 13JUL2000

This document contains a summary of website content requirements and restrictions for publicly accessible Navy websites. A website satisfies the definition of being "publicly accessible" if any of the content on the website is accessible by the public via anonymous access. Restricting access by domain validation or SSL without client-side authentication is not sufficient to be excluded from the definition of "publicly accessible"

Authorized publicly accessible web presence:

- No entity below the command level or its' equivalent is authorized to establish a publicly accessible web site.

  [Ref B, encl 2: 1.c]

  Only commissioned units are authorized to register a domain name for a website. Non-commands are allowed to create a web presence but only as a sub-web off of an authorized web site. Sub-webs will appear as an integral part of their command level parent web site. For instance, sub-webs will be implemented with the same "theme" as the parent

web site and any "home" buttons on the sub-web pages must link to the parent's web site home page only.

Navy publicly accessible web sites MUST:

- **Contain the Full command's organizational name.**

  [Ref B, encl 2: 2.b.1]

  The full command organizational name (with no abbreviations) must be prominently displayed on the web site home page.

- **Contain the statement "This is an official U.S. Navy web site".**

  [Ref B, encl 2: 2.b.2]

  The exact phrase "This is an official U.S. Navy web site" must be prominently displayed on the web site home page.

- **Contain a tailored Privacy and Security Notice.**

  [Ref B, encl 2: 2.b.3; Ref A part V, 4]

  The web site Privacy and Security Notice or a hyperlink to the web site Privacy and Security Notice must be prominently displayed on the web site home page.

  The Privacy and Security Notice MUST BE verbatim from Refs A or B. The only authorized modifications are to substitute the command's organizational name in the places indicated.

  Privacy and Security Notice example per Ref B:

  "Notice: This is a U.S. Government Web Site

  1. This is a World Wide Web site for official information about [the name of command/activity]. It is provided as a public service by [command/activity name and servicing command if applicable]. The purpose is to provide information and news about the [name of command/activity] to the general public.
  2. All information on this site is public domain and may be distributed or copied unless otherwise specified. Use of appropriate byline/photo/image credits is requested.
  3. Unauthorized attempts to upload information or change information on this Web site are strictly prohibited and may be punishable under the Computer Fraud and Abuse Act of 1986 and the National Information Infrastructure Protection Act.
  4. For site security purposes and to ensure that this service remains available to all users, this government computer system employs software programs to monitor network traffic to identify unauthorized attempts to upload or change information, or otherwise cause damage.
  5. Except for authorized law enforcement investigation and to maintain required correspondence files, no other attempts are made to identify individual users or their usage habits. Raw data logs are used to simply determine how many users are accessing the site, which pages are the most popular, and, from time to time,

from which top level domain users are coming. This data is scheduled for regular destruction in accordance with National Archives and Records Administration guidelines."

- Contain the webmaster contact information.

  [Ref B, 7.d.4]

  Information on how to contact the webmaster must be displayed on the web site home page or at least contained within the source code of the home page. Ideally webmaster contact information should be listed on the web site home page and should include: an e-mail address, work telephone number and work mailing address.

- **Contain a link to parent command or Immediate Superior in Chain (ISIC).**

  [Ref B, encl 2: 2.c.2]

- **Contain a link to the official U.S. Navy web site: .**

  [Ref B, encl 2: 2.c.1]

- **Contain a link to Navy recruiting web site: .**

  [Ref B, encl 2: 2.c.3]

- External links to non U.S. Government web sites must be accompanied by a disclaimer statement.

  [Ref A, part II, 8.2]

  External links to non-government web sites that directly support the command's mission are authorized but a disclaimer statement must be displayed on the page or pages listing external links or through an intermediate "exit notice" page.

  External link disclaimer notice Example:

  "The appearance of external hyperlinks does not constitute endorsement by the United States Department of Defense, the United States Department of the Navy and [command name] of the linked web sites, or the information, products or services contained therein. For other than authorized activities such as military exchanges and Morale, Welfare and Recreation (MWR) sites, the United States Department of Defense, the Department of the Navy and [command name] does not exercise any editorial control over the information you may find at these locations. Such links are provided consistent with the stated purpose of this DoD web site."

- **All solicitations from the web site visitor must be accompanied by a Privacy Advisory**.

  [Ref B, encl 2: 5.d; Ref A part II, 12.2]

  The term "solicitation" encompasses any and all requests for submissions including surveys, forms, and webmaster feedback.

  Privacy Advisory example:

  "We will not obtain personally identifying information about you when you visit our site unless you choose to provide such information to us. If you choose to send email to the site webmaster or submit an online feedback form, any contact information that you provide will be solely used to respond to your request and not stored."

- **Have the written approval of SECDEF for the use of persistent cookies.**

  [Reference A, Part II, 12.3.2]

  A cookie that is set to expire greater than 24 hours after being set is considered to be "persistent".

- **All session cookies and pre-approved persistent cookies must be accompanied by a disclosure statement.**

  [Ref A, partII, 12.3.1]

  The disclosure statement must state:

  - that the site contains a cookie,
  - why the cookie is being used,
  - the safeguards in place to protect any information collected.

- **A Notice and Consent Banner.**

  [Ref A, part V, 4.2]

  A verbatim Notice and Consent Banner (sometimes referred to as a DoD Warning Banner) must be prominently displayed at the access point for web sites where access is controlled by a level 3 Security and Access Control mechanism (i.e., User authentication).

  Notice and Consent Banner Notice Example:

  "This is a Department of Defense Computer System. This computer system, including all related equipment, networks, and network devices (specifically including Internet access) are provided only for authorized U.S. Government use. DoD computer systems may be monitored for all lawful purposes, including to ensure that their use is authorized, for management of the system, to facilitate protection against unauthorized access, and to verify security procedures, survivability, and operational security. Monitoring includes active attacks by authorized DoD entities to test or verify the security of this system. During monitoring, information may be examined, recorded, copied and used for authorized purposes. All information, including personal information, placed or sent over this system may be monitored. Use of this DoD computer system, authorized or unauthorized, constitutes consent

to monitoring of this system. Unauthorized use may subject you to criminal prosecution. Evidence of unauthorized use collected during monitoring may be used for administrative, criminal, or other adverse action. Use of this system constitutes consent to monitoring for these purposes."

Navy publicly accessible web sites must NOT contain:

- Overt warning signs, or words of warning or danger in association with the Privacy and Security Notice. The Privacy and Security Notice can only be identified with the phrase "Privacy and Security Notice".

  [Ref A, part II, 7; Ref B, encl 2: 2.b.3]

  Indicators that create a misperception of danger in association with the Privacy and Security Notice will not be used. The Privacy and Security Notice can only be identified with the phrase "Privacy and Security Notice".

- Altered photos (other than standard photographic processes).

  [Ref B, encl 2: 3.b]

  Some alterations are acceptable as long as the alterations do not defer from the original intent.

- FOUO or above information.

  [Ref A, part V, 2.; Ref B, encl 2: 3.c.3]

- **Personally identifying content.**

  [Ref A, Part V, 2.2; Ref B, encl 2: 3.c.2, 2:3.d.2; Ref D]

  Any information that can be used to identify DoD individuals. Exception: Command Executives (i.e., CO, XO, CMC) can be identified by photo and name only. The following table lists specific information that is not to be divulged:

  - Social Security Number
  - Marital Status
  - Age
  - Home address or phone numbers
  - Birth date or place
  - Family members
  - Race, religion, citizenship
  - City home of record

- Personalized e-mail address

- Proprietary or copyrighted content.

  [Ref A, Part V, 2.3; Ref B, encl 2: 3.c.5, 3.d.5]

- Operational Lessons Learned.

  [Ref A, Part II, 3.5.3.1; Ref B, encl 2: 3.d.1]

- Information revealing sensitive military operations, exercises, vulnerabilities, maps identifying command and operational facilities.

  [Ref A: part II, 3.5.3.1, 3.5.3.2, Part V, 2.1; Ref B, encl 2: 3.d.1]

- **Information for specialized, internal audience or of questionable value to the general public that is not access limited by at least domain restriction.**

  [Ref A, Part I, 4.3.1, Part II, Part V, 3; Ref B, encl 2: 3.d.3]

  Only content that is specifically targeted for the general public should be posted on web sites that have no access restrictions implemented. Content intended for an internal audience will, at a minimum, have access limited by domain restriction.

- **Information that places national security, personnel, assets, or mission effectiveness at unacceptable risk.**

  [Ref A, part II, 3.6.2, part V, 2.; Ref B, encl 2: 3.d.1]

- **Phone numbers that can be associated with individuals. Only phone numbers for commonly requested resources and services or for office codes are allowed.**

  [Ref D]

- Product endorsements, preferential treatment of any private organization or product, or references including logo or text indicating that the site is “best viewed” with any specific web browsers.

  [Ref A, part II, 3.5.6, 8.1.2, 8.1.4; Ref B, encl 2: 3.d.4]

- Contain links or references to documents within DoD Web sites that have security and access controls.

  [Ref A, Part II, 3.6.3]

However, it is permissible to link to log-on sites, provided details as to the controlled site's contents are not revealed.

- Content duplicated from other military web resources.

  [Ref A, Part II, 2.3; Ref B, encl 2: 3.d.9]

  Navy web sites may reference (via hyperlink) these external resources instead.

  For example you may provide a link to:

  http://www.chinfo.navy.mil/navpalib/factfile/ffiletop.html for ship characteristics.

Naval IW commands include the Naval Information Warfare Activity (NIWA) and the Fleet Information Warfare Center (FIWC).

FIWC Electronic Warfare Reprogramming Facilities are located in Chesapeake, Virginia and Honolulu.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix F
TopSpeed Driver Supported Features

TopSpeed Driver – Supported Features

File Attributes	Supported
Create	Yes
Driver (filetype)	Yes
Name	Yes
Encrypt	Yes
Owner (Password)	Yes
Reclaim	No
Prefix	Yes
Bindable	Yes
Thread	Yes
External (member)	Yes
DLL (flag)	Yes
OEM	Yes

File Structures	Supported
Index	Yes
Key	Yes
Memo	Yes
BLOB	Yes
Record	Yes

Index, key, memo.	Supported
Binary	Yes
Dup	Yes
NoCase	Yes
Opt	Yes
Primary	Yes
Name	Yes
Ascending Components	Yes
Descending Components	Yes
Mixed Components	Yes

Fixed Attributes	Supported
Dim	Yes
Over	Yes
Name	Yes

File Procedures	Supported
BOF (file)	Yes
Pack (file)	Yes
Pointer (file)	Yes
Pointer (key)	Yes
Position (file)	Yes
Records (file)	Yes
Records (key)	Yes
Remove (file)	Yes
Rename (file)	Yes
Send (file, message)	Yes
Share (file)	Yes
Status (file)	Yes
Stream (file)	Yes
Unlock (file)	Yes

Record Access	Supported
Add (file)	Yes
Add (file, length)	No
Append (file)	Yes
Append (file, length)	No
Delete (file)	Yes
Get (file, key)	Yes
Get (file, filePointer)	Yes
Get (file, filePointer, length)	No
Get (file, keyPointer)	Yes
Hold (file)	Yes
Next (file)	Yes
NoMemo (file)	Yes
Previous (file)	Yes
Put (file)	Yes
Put (file, filePointer)	Yes
Put (file, filePointer, length)	No
Release (file)	Yes
ReGet (file, string)	Yes
ReGet (key, string)	Yes
ReSet (file, string)	Yes
ReSet (key, string)	Yes
Set (file)	Yes
Set (file, key)	Yes

TopSpeed Driver – Supported Features

File Procedures	Supported
Buffer (file)	No
Build (file)	Yes
Build (key)	Yes
Build (index)	Yes
Build (index, components)	Yes
Build (index, comp., filter)	Yes
Bytes (file)	Yes
Close (file)	Yes
Copy (file)	Yes
Create (file)	Yes
Duplicate (file)	Yes
Duplicate (key)	Yes
Empty (file)	Yes
EOF (file)	Yes
Flush (file)	Yes
Lock (file)	Yes
Name (label)	Yes
Open (file, mode)	Yes

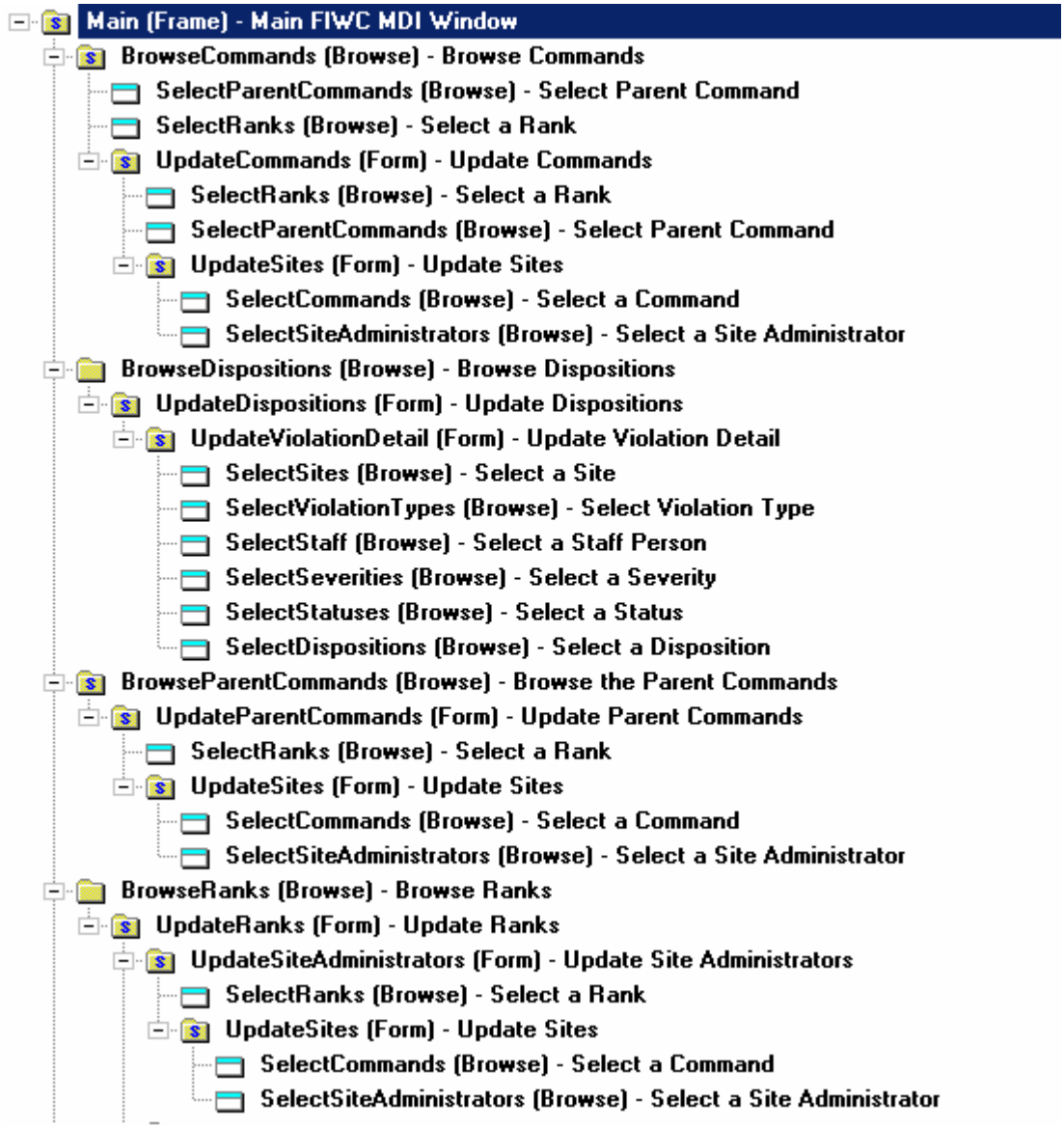
Record Access	Supported
Set (file, filePointer)	Yes
Set (key)	Yes
Set (key, key)	Yes
Set (key, keyPointer)	Yes
Set (key, key, filePointer)	Yes
Skip (file, count)	Yes
Watch (file)	Yes

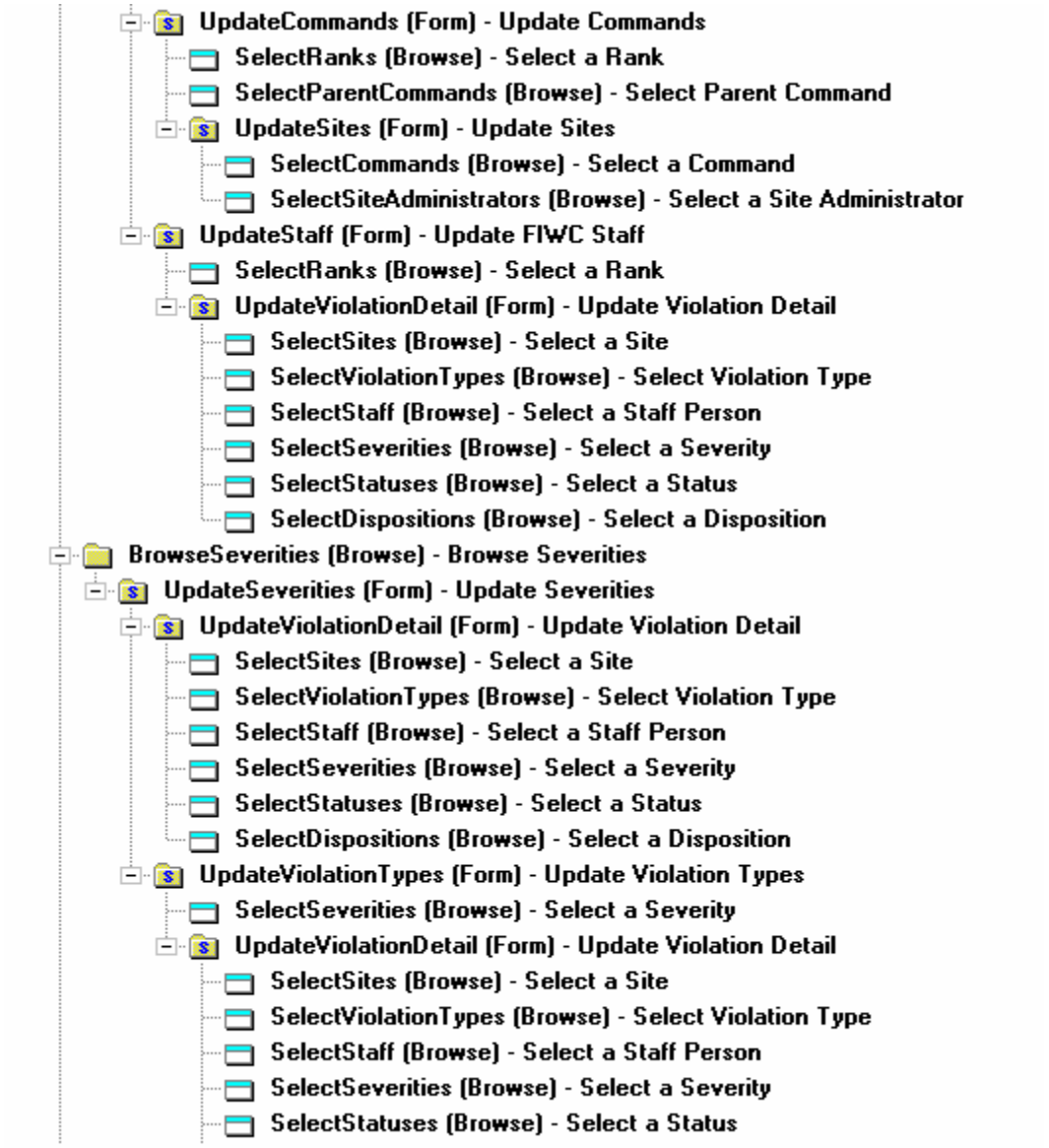
Transaction Processing	Supported
Logout (timeout, file, …)	Yes
Commit	Yes
RollBack	Yes

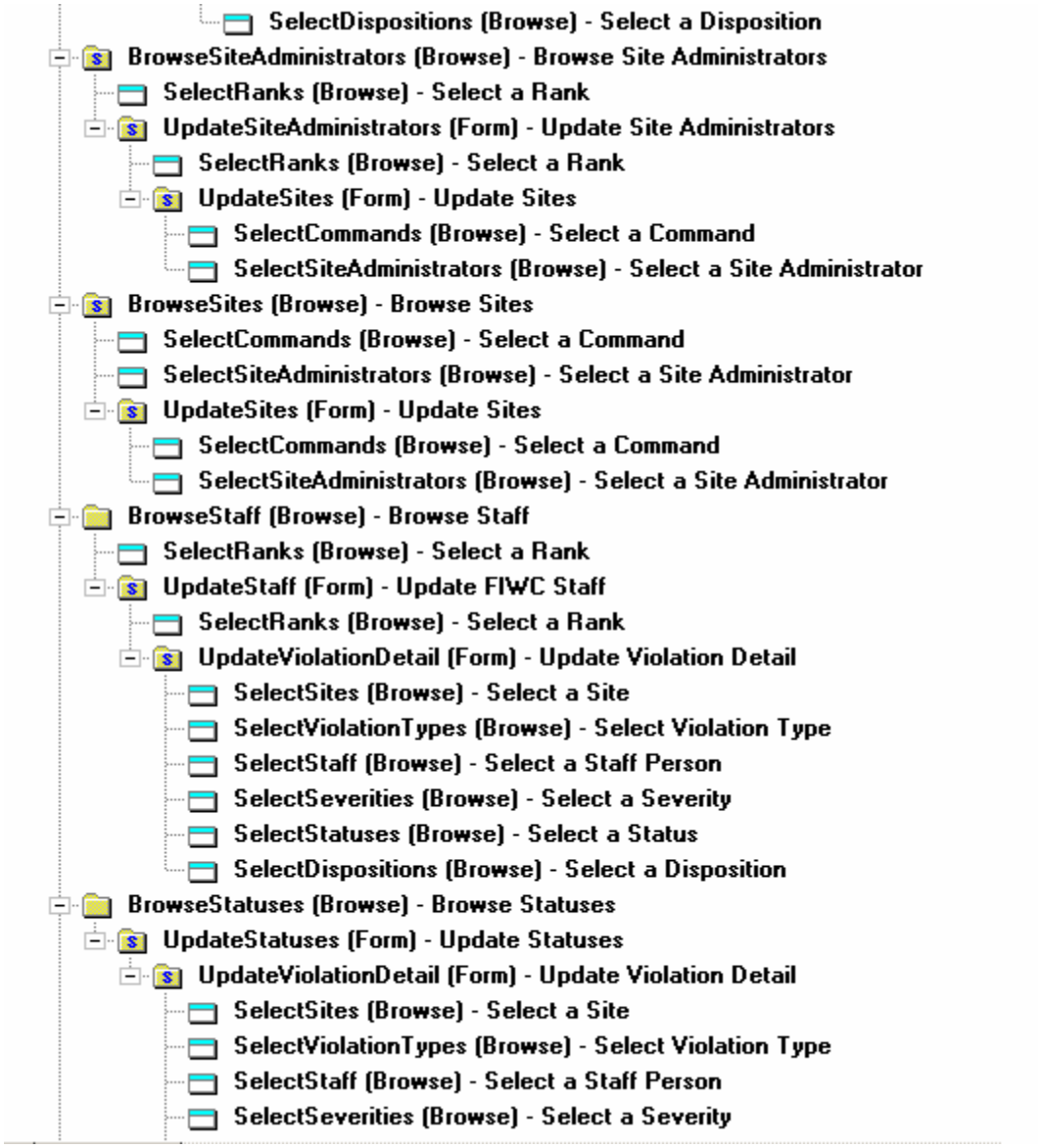
Null Data Processing	Supported
Null (field)	Yes
SetNull (field)	Yes
SetNonNull (field)	Yes

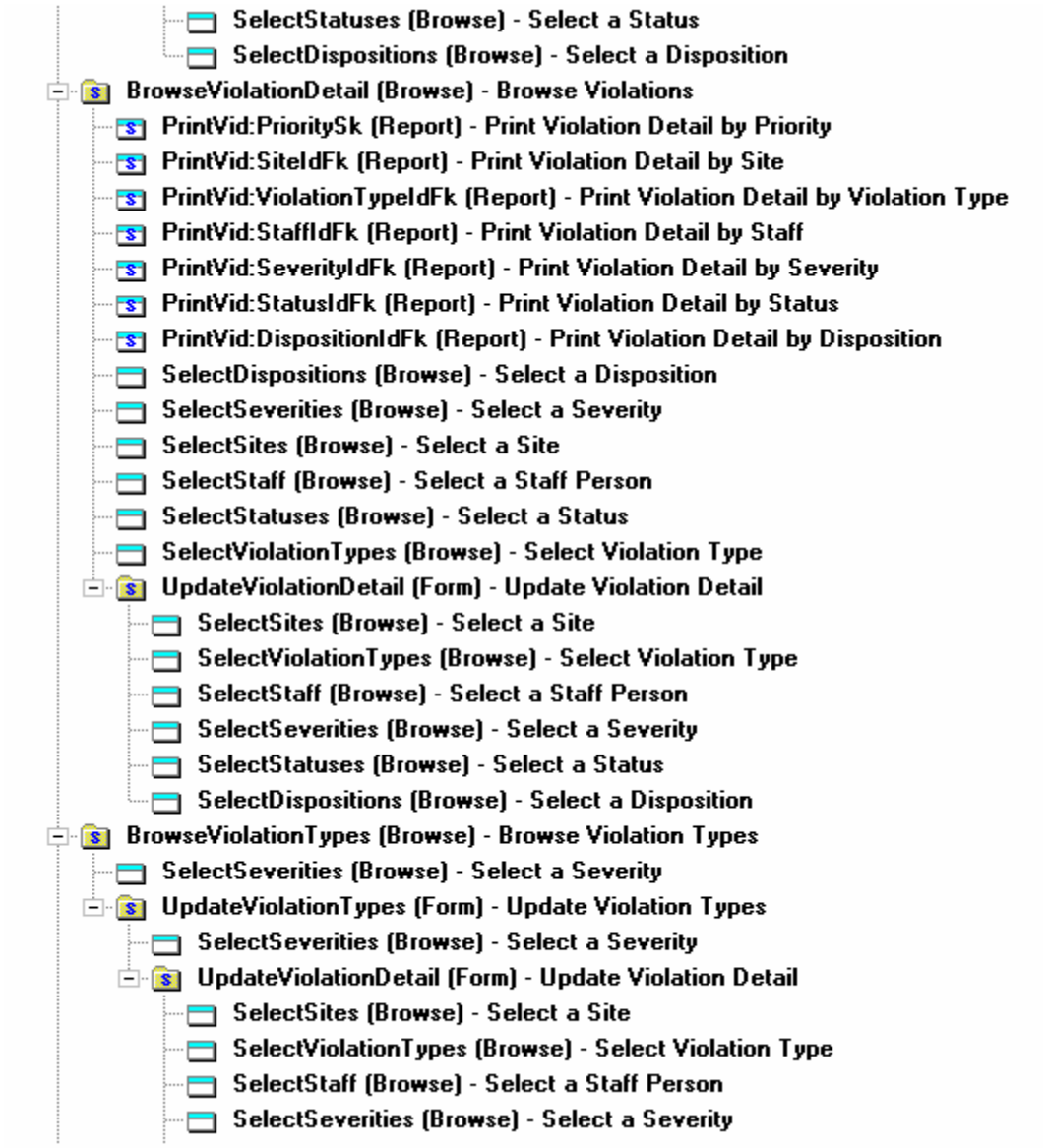
Appendix G
Procedural Tree Chart

The following pages depict the FIWC Website Compliance Application procedures in a hierarchical context. Procedures invoked by multiple parent procedures occur multiple times, being depicted as many times as they are invoked.




UpdateCommands (Form) - Update Commands
SelectRanks (Browse) - Select a Rank
SelectParentCommands (Browse) - Select Parent Command
UpdateSites (Form) - Update Sites
SelectCommands (Browse) - Select a Command
SelectSiteAdministrators (Browse) - Select a Site Administrator
UpdateStaff (Form) - Update FIWC Staff
SelectRanks (Browse) - Select a Rank
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity
SelectStatuses (Browse) - Select a Status
SelectDispositions (Browse) - Select a Disposition
BrowseSeverities (Browse) - Browse Severities
UpdateSeverities (Form) - Update Severities
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity
SelectStatuses (Browse) - Select a Status
SelectDispositions (Browse) - Select a Disposition
UpdateViolationTypes (Form) - Update Violation Types
SelectSeverities (Browse) - Select a Severity
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity
SelectStatuses (Browse) - Select a Status


SelectDispositions (Browse) - Select a Disposition
BrowseSiteAdministrators (Browse) - Browse Site Administrators
SelectRanks (Browse) - Select a Rank
UpdateSiteAdministrators (Form) - Update Site Administrators
SelectRanks (Browse) - Select a Rank
UpdateSites (Form) - Update Sites
SelectCommands (Browse) - Select a Command
SelectSiteAdministrators (Browse) - Select a Site Administrator
BrowseSites (Browse) - Browse Sites
SelectCommands (Browse) - Select a Command
SelectSiteAdministrators (Browse) - Select a Site Administrator
UpdateSites (Form) - Update Sites
SelectCommands (Browse) - Select a Command
SelectSiteAdministrators (Browse) - Select a Site Administrator
BrowseStaff (Browse) - Browse Staff
SelectRanks (Browse) - Select a Rank
UpdateStaff (Form) - Update FIWC Staff
SelectRanks (Browse) - Select a Rank
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity
SelectStatuses (Browse) - Select a Status
SelectDispositions (Browse) - Select a Disposition
BrowseStatuses (Browse) - Browse Statuses
UpdateStatuses (Form) - Update Statuses
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity


SelectStatuses (Browse) - Select a Status
SelectDispositions (Browse) - Select a Disposition
BrowseViolationDetail (Browse) - Browse Violations
PrintVid:PrioritySk (Report) - Print Violation Detail by Priority
PrintVid:SiteIdFk (Report) - Print Violation Detail by Site
PrintVid:ViolationTypeIdFk (Report) - Print Violation Detail by Violation Type
PrintVid:StaffIdFk (Report) - Print Violation Detail by Staff
PrintVid:SeverityIdFk (Report) - Print Violation Detail by Severity
PrintVid:StatusIdFk (Report) - Print Violation Detail by Status
PrintVid:DispositionIdFk (Report) - Print Violation Detail by Disposition
SelectDispositions (Browse) - Select a Disposition
SelectSeverities (Browse) - Select a Severity
SelectSites (Browse) - Select a Site
SelectStaff (Browse) - Select a Staff Person
SelectStatuses (Browse) - Select a Status
SelectViolationTypes (Browse) - Select Violation Type
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity
SelectStatuses (Browse) - Select a Status
SelectDispositions (Browse) - Select a Disposition
BrowseViolationTypes (Browse) - Browse Violation Types
SelectSeverities (Browse) - Select a Severity
UpdateViolationTypes (Form) - Update Violation Types
SelectSeverities (Browse) - Select a Severity
UpdateViolationDetail (Form) - Update Violation Detail
SelectSites (Browse) - Select a Site
SelectViolationTypes (Browse) - Select Violation Type
SelectStaff (Browse) - Select a Staff Person
SelectSeverities (Browse) - Select a Severity

With respect to the apparent hierarchical "orphaning" of the last four procedures: The Clarion hierarchical procedure view is based upon formal procedural links. Programmatic invocations (calls) are transparent to this view. The last four procedures – Print Violation Detail by Violation Identifier, Update Passwords, Browse Passwords, and Login – are all invoked programmatically rather than being invoked formally. This gives them the appearance in this presentation of being orphaned.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix H
Embedded Logic

The following pages contain the more significant sections of program code embedded in the prototype application's procedures. The purpose of each block of code is conveyed in a comment line heading the embed. Clarion comments are set off by a leading exclamation point (!). There is other custom code, not represented here, which performs routine tasks such as actuating drop lists, priming attributes upon tuple insertion, calling reports and select procedures, and linking table attributes to subset list boxes.

Procedure: Main
Embed Point: Accept Loop, After CASE EVENT() Handling
Embed Code:

```
!--- Display Date & Time in Status Bar ---
AppFrame {Prop:StatusText,2} = Clip
(Loc:DayText[(Today()%7)+1]) & ', ' & Format(Today(),@D4)
AppFrame {Prop:StatusText,3} = Glo:PrivilegeLevel
AppFrame {Prop:StatusText,4} = 'Activity: ' & Clip
(Format(Clock(),@T3))
Display
```

Procedure: Main
Embed Point: Control Event Handling After Generated Code
Embed Code:

```
! --- Filter Supervisory Access Only ---
If Glo:PrivilegeLevel = 'Supervisory'
  Start (BrowsePasswords,50000)
Else
  Message ('User authorization precludes
            access','Disallowed',Icon:HAND,Button:Ok)
End
```

Procedure: Login
Embed Point: Control Event Handling After Generated Code
Embed Code:

```
!--- Process StaffId and Password ---
  CheckOpen (Chains,1)
! Message
('UserId:',Glo:StaffId,Icon:Exclamation,Button:Ok)
! Message
('Password:',Glo:Password,Icon:Exclamation,Button:Ok)
  Pwd:StaffId = Glo:StaffId
  Get (Chains,Pwd:StaffIdPk)
  If ErrorCode ()
    Message (Error (),'Unregistered User -
             Try again',Icon:Exclamation,Button:Ok)
    Select (?Glo:StaffId)
  ElsIf Pwd:Password = Glo:Password
    Glo:StaffId = Pwd:StaffId
    Glo:PrivilegeLevel = Pwd:PrivilegeLevel
    Unhide (?Pwd:PrivilegeLevel)
    Display (?Pwd:PrivilegeLevel)
    Return
  Else
    Loop
      Case Message ('Invalid Password -
           Exit?','ERROR!',,BUTTON:Yes+BUTTON:No,BUTTON:No)
      Of Button:No
        Select (?Glo:Password)
      Break
      Of Button:Yes
        Halt ()
      End
    End
  End
```

Procedure: BrowseViolationDetail
Embed Point: Format an Element on the Browse Queue
Embed Code:

```
!-------------------- Load Variables --------------------
! Indirect chain: ViolationDetail -> Sites -> Commands

Sma:SiteId = Vid:SiteId
Get (Sites,Sma:SiteIdPk)
Com:CommandId = Sma:CommandId
Get (Commands,Com:CommandIdPk)
Display (?Com:CommandName)
Display (?Com:OfficerInCharge)

Sta:StatusId = Vid:StatusId
Get (Statuses,Sta:StatusIdPk)
Display (?Sta:Description)
Display (?Vid:StatusDate)

Dsp:DispositionId = Vid:DispositionId
Get (Dispositions,Dsp:DispositionIdPk)
Display (?Dsp:Description)
Display (?Vid:DispositionDate)

Stf:StaffId = Vid:StaffId
Get (Staff,Stf:StaffIdPk)
Loc:FullName = Clip (Stf:LastName) & ', ' & (Stf:FirstName)
Display (?Loc:FullName)

Sev:SeverityId = Vid:SeverityId
Get (Severities,Sev:SeverityIdPk)
Display (?Sev:Description)

Vit:ViolationTypeId = Vid:ViolationTypeId
Get (VioTypes,Vit:ViolationTypeIdPk)
Display (?Vit:Description)
Display (?Vid:Priority)
Display (?Vid:Remarks)
```

Procedure: UpdateCommands
Embed Point: Before Call to RIUpdate if Record Changed
Embed Code:

```
! ------- Duplicate Modifications in Parent Command -------

Par:CommandId = Com:CommandId
Par:CommandName = Com:CommandName
Get (ParentCommands,Par:CommandIdPk)

! Message ('Injecting ParentCommand
record','Debug',Icon:Application,Button:Ok)
Add (ParentCommands)                ! Create skeleton
entry
Par:CommandId = Com:CommandId
Par:CommandName = Com:CommandName
Par:CommandLevel = Com:CommandLevel
Par:OfficerInCharge = Com:OfficerInCharge
Par:DirectedMessageRecipient = Com:DirectedMessageRecipient
Par:RankId = Com:RankId
Par:ReviewSchedule = Com:ReviewSchedule
Par:ActiveDate = Com:ActiveDate
Par:InactiveDate = Com:InactiveDate
Par:Remarks = Com:Remarks
Put (ParentCommands)                ! Flush buffer
```

Procedure: UpdateCommands
Embed Point: Beginning of Procedure After Opening Files
Embed Code:

```
!-- Populate Externally Linked Descriptions --

LocalCommandId = Com:CommandId    ! Store Current Command

If OriginalRequest = InsertRecord
!  Do nothing - field priming takes care of it
Else
   Par:CommandId = Com:ParentCommandId
   Get (ParentCommands,Par:CommandIdPk)
   Display (?Par:CommandName)
   Rnk:RankId = Com:RankId
   Get (Ranks,Rnk:RankIdPk)
   Display (?Rnk:Description)
End

Display (?Com:ParentCommandId)
Display (?Com:RankId)
```

Procedure: UpdateCommands
Embed Point: When Completed Before Writing to Disk
Embed Code:

```
! --- Duplicate Insertions to Parent Command ---

If LocalRequest = InsertRecord
   Par:CommandId = Com:CommandId
   Par:CommandName = Com:CommandName
   Get (ParentCommands,Par:CommandIdPk)
   Message ('Inserting ParentCommand
   record','Debug',Icon:Application,Button:Ok)
   Add (ParentCommands)                    ! Create skeleton
entry
   Par:CommandId = Com:CommandId
   Par:CommandName = Com:CommandName
   Par:CommandLevel = Com:CommandLevel
   Par:OfficerInCharge = Com:OfficerInCharge
   Par:DirectedMessageRecipient =
Com:DirectedMessageRecipient
   Par:RankId = Com:RankId
   Par:ReviewSchedule = Com:ReviewSchedule
   Par:ActiveDate = Com:ActiveDate
   Par:InactiveDate = Com:InactiveDate
   Par:Remarks = Com:Remarks
   Put (ParentCommands)                    ! Flush buffer
End
```

Procedure: UpdateParentCommands
Embed Point: Beginning of Procedure After Opening Files
Embed Code:

```
!--- Populate Externally Linked Descriptions ---

LocalCommandId = Par:CommandId           ! Store Current
Command

If OriginalRequest = InsertRecord
!  Do nothing - field priming takes care of it
Else
   Rnk:RankId = Par:RankId
   Get (Ranks,Rnk:RankIdPk)
   Par:RankId = Rnk:RankId
   Display (?Rnk:Description)
End

Display (?Par:RankId)
```

Procedure: UpdateViolationDetail
Embed Point: Beginning of Procedure After Opening Files
Embed Code:

```
!--- Populate Externally Linked Descriptions ---

If OriginalRequest = InsertRecord
!  Do nothing - field priming takes care of it
!  Clear ()
Else
   Sma:SiteId = Vid:SiteId
   Get (Sites,Sma:SiteIdPk)
   Vid:SiteId = Sma:SiteId
   Vit:ViolationTypeId = Vid:ViolationTypeId
   Get (VioTypes,Vit:ViolationTypeIdPk)
   Vid:ViolationTypeId = Vit:ViolationTypeId
   Stf:StaffId = Vid:StaffId
   Get (Staff,Stf:StaffIdPk)
   Vid:StaffId = Stf:StaffId
   Loc:FullName = Clip (Stf:LastName) & ', ' &
                  Clip( Stf:FirstName)
   Sev:SeverityId = Vid:SeverityId
   Get (Severities,Sev:SeverityIdPk)
   Vid:SeverityId = Sev:SeverityId
   Sta:StatusId = Vid:StatusId
   Get (Statuses,Sta:StatusIdPk)
   Vid:StatusId = Sta:StatusId
   Dsp:DispositionId = Vid:DispositionId
   Get (Dispositions,Dsp:DispositionIdPk)
   Vid:DispositionId = Dsp:DispositionId
   Display (?Vid:SiteId)
   Display (?Sma:MainSiteURL)
   Display (?Vid:ViolationTypeId)
   Display (?Vit:Description)
   Display (?Vid:ViolationTypeId)
   Display (?Loc:FullName)
   Display (?Vid:SeverityId)
   Display (?Sev:Description)
   Display (?Vid:StatusId)
   Display (?Sta:Description)
   Display (?Vid:DispositionId)
   Display (?Dsp:Description)
End
```

Embed Code:

```
!--- Store Loc:ViolationId ---
Loc:ViolationId = Vid:ViolationId
```

Appendix I

Web Quality Central Data Sheet Abstract

COAST™ Web Quality Central

Centralize and automate your post-deployment Web quality management

COAST Web Quality Central delivers a powerful server-based solution that provides enterprises with an automated, centralized solution for Web quality management and site verification. Based on award-winning technology, COAST Web Quality Central offers integrated database support, extremely customizable scripting and reporting capabilities and unlimited monitor events. Since 1996, COAST Software has been focused on developing the best Web quality management solutions on the market. Always evolving to support new Web technology, COAST Web Quality ® Central supports dynamic Web pages, Microsoft .NET Web Services testing, JavaScript navigation, JavaServer pages, ® Macromedia Flash, Cookies and Session IDs .

Using COAST Web Quality Central, Web stakeholders will regain control and confidence over the performance of their site. Our comprehensive scans highlight areas for improvement and deliver tailored reports that can serve as an archive of the changes and developments of a Web site over time. Advanced PageRules™ allow Web teams to easily determine whether their Web site conforms to all corporate standards such as copyright and privacy notices and will help perform accessibility testing using Section 508 or W3C specifications as a standard.

We offer a simple, uncomplicated licensing program that provides organizations with a compelling business case: COAST Software performs the critical function of Web quality management more efficiently and effectively than manual-testing methods enabling Web resources to be deployed more productively while providing tailored reports that deliver the information needed to maintain and develop a successful Web site.

Compelling ROI Ensures conformance to corporate standards

COAST Web Quality Central conducts comprehensive, automated scans of your Web site allowing you to avoid inefficient and costly manual testing and ensuring your business-critical Web site delivers a positive customer experience.

Verify that your Web site meets your organization's corporate standards for elements like copyright, legal notices, and other common look-and-feel elements. The PageRules™ feature in COAST Web Quality Central allows you to write your own compound test rules confirming the absence or presence of specific display text, tag text, links, applets, controls, scripting, forms and frames generated by your code.

Advanced reporting with database integration Integrates with content management workflows

Database integration delivers unlimited reporting flexibility and the capability to drill down into your Web site data to deliver powerful, meaningful reports and trend analysis.

COAST Web Quality Central automates the content quality management task by ® integrating with in-house content management programs like Interwoven TM ® TeamSite and Microsoft CMS. COAST Web Quality Central automatically verifies content against a user-defined set of rules and returns a pass/fail report to the author or QA team.

- Flexible reporting capabilities with database integration
- Provides a robust platform that will scale to meet the future demands of your enterprise Web site
- Powerful site scripting capabilities for verifying transactions and online forms
- Dramatically reduces the time and cost associated with manual Web quality management
- Ensures your Web site conforms to organizational standards Allows multiple users to verify Web content
- Integrates with your current content management workflow
- Automates your current Web site verification tasks
- Identifies potential Web site accessibility issues
- Monitors your Web site 24/7
- Allows multiple users to verify Web site content
- Accessibility testing
- Manages, maintains and inventories your Web site
- Quickly & automatically locates Web site errors
- Provides data for Web site archives
- Ensures no objectionable pointers
- Powerful site scripting capabilities
- Visitor analysis
- Integration with Rational® Software

Appendix J
Selected Correspondence
Email and Newsgroup TPS File Encryption Threads

----- Original Message -----
From: "Vickie" <Seahorse@Redshift.com>
To: <clarion@attryde.com>
Sent: Saturday, May 31, 2003 20:14
Subject: Clarion encryption

Hi,

I have a "proof of concept" prototype (good start on a production application) for a thesis. The thesis is for the US Navy command. The app and thesis have to do with policing USN website conformance to certain government regulations.

I am using the encrypt feature on a password table, but cannot for the life of me find out what encryption algorithm Clarion uses. Do you know?

Thanks,

Vickie G.

----- Original Message -----
From: "Paul Attryde" <paul@attryde.com>
To: "Vickie" <Seahorse@Redshift.com>
Sent: Sunday, June 01, 2003 1:17 PM
Subject: Re: Clarion encryption

Assuming your talking about a .TPS file, it's a proprietary method that (as far as I know) isn't documented anywhere. It can be broken with time, and there are people out there that have done it, so depending on what it is that your storing you may want to add some additional steps. There's a Clarion implementation of MD5 floating around on the 'net, I'm sure you could find it if you needed to.

HTH, Paul

---

2 June 2003

----- Original Message -----
From: "Ben E. Brady" <support@clariondeveloper.com>
To: "Vickie" <Seahorse@Redshift.com>
Sent: Sunday, June 01, 2003 11:09 PM
Subject: Re[2]: Web Site Contact

Vickie,

Just for your edification, our Secure Address Book product (www.firewallreporting.com/sab) relies on the MD5 algorithm to protect the address book from being exploited by email worms and viruses.

Essentially it works like this...

Using a TPS encrypted database (a proprietary 512 bit encryption algorithm) the user is prompted for a password prior to the database creation. This password is stored in a global memory variable for a very brief period of time.

Due to a fundamental flaw in Clarion (which allows the password of an encrypted TPS file to be extracted from a Clarion produced executable program) most Clarion developers simply store the password of the encrypted database in the data dictionary. The appropriate way to do this is to assign a variable to the encryption password and then populate that variable at run time so as to eliminate the problem.

I devised the MD5 algorithm implementation to take the password as entered by the user and produce an MD5 signature which is then used to actually create the 32 byte (128 bit) password which is then used to create the encrypted database. The MD5 algorithm provides the appropriate level of case sensitivity as well as ensuring that an absolutely unique signature is provided.

The resulting password used for the TPS encryption cannot be reverse engineered as the MD5 algorithm is a 'one way' encryption.

If one were to look at the Clarion executable they MIGHT be able to determine what the name of the variable is that would contain the password, however, they would have to know EXACTLY where to look in the executable binary code. In my applications this is further obfuscated by the use of an executable 'packer' which actually compresses the executable (usually making it approximately 60 percent smaller in the process due to the inefficiencies of the Windows Program Executable file format) and obscures the data locations of variable names and other structure identifiers.

There is reportedly one person that I have heard of that can read encrypted TPS files (to my knowledge he must have access to the executable and data file and the executable must be running in memory). Were he to only have the TPS file itself he has stated to me that he would not be able to read it. This indicates to me that he has not actually 'cracked' the TPS encryption model.

Hope this serves to clear up any questions you might have regarding the use of TPS encrypted files.

--
Best regards,
Ben E. Brady
Brady & Associates, LLC.

Ben,

That's awesome. Great information. I will use it all. I am amazed that Clarion didn't obscure the password mapping in the executable.

I briefly looked at your address book solution and wondered whether you had used MD5 in it.

Thank you very much.

Vickie

Victoria Galante
vjgalant@nps.navy.mil
831/372-3748
U.S. Cyber-Corps SFS Program
Naval Postgraduate School

Hi Vickie;

An org chart can certainly be done with a tree structure. For example:

```
-Bill (president)
  -Fred(VP Engineering)
    -Sally(QA)
      -Mildred(Head Checker)
        Joyce(Grunt Checker)
        Larry(Grunt Checker)
      -Jerry(Programming Mgr)
        -Carl(Systems Analyst)
          John(Programmer III)
          Cindy(Programmer II)
          Ralph(Programmer II)
      -Lisa(Hardware Mgr)
          Bilbo(Eng Tech I)
          Sam(Eng Tech II)
          -Sue(Documentation Specialist)
            Renate(Typist)
            Tyrone(Typist)
          Mr. Fixit(WhateverNeedsToBeDone)
        .
        .
        .
      etc
```

The tree structure is merely a "set theory" mathematical representation ... per above

Key Fields:

```
1     2      3       4         5

Bill |      |       |         |
Bill | Fred |       |         |
Bill | Fred | Sally |         |
Bill | Fred | Sally | Mildred |
Bill | Fred | Sally | Mildred | Joyce
Bill | Fred | Sally | Mildred | Larry
Bill | Fred | Jerry |         |
Bill | Fred | Jerry | Carl    |
Bill | Fred | Jerry | Carl    | John
Bill | Fred | Jerry | Carl    | Cindy
Bill | Fred | Jerry | Carl    | Ralph
Bill | Fred | Lisa  |         |
Bill | Fred | Lisa  | Bilbo   |
Bill | Fred | Lisa  | MrFixIt |
Bill | Fred | Lisa  | Sam     |
               ...
```

> The problem is, not how you get from Bill to Cindy who works in Carl's department, but how do you get from Cindy to who she works for? The typical hierarchical representation only takes you one way, i.e., from the top down. This is the problem with all hierarchical structures. How do you find a specific item? Search.
>
> In the record layout, you need a backward pointer ... i.e., in Cindy's record layout, a text field that points to Carl's record. A program must be able to traverse up the chain as well as down. The question "Who is this person's manager at level-2 in the organization?" is an iterative process of tracing back up the tree.
>
> Make sense? I hope so ... it's getting late and not much is making sense.
>
> Greg

If you use the approach I was talking about, having multiple keys, 1 key for each level, going down the tree is just as easy as going up the tree. Not only that, you can have multiple tabs representing multiple sort orders for each level of command.

I agree with you that a master file and a relationship file is needed. Keep it simple! The KIS method always produces a cleaner solution!

It means rework, and this is a thesis project, but it looks like my straight line. I've already wasted about 14 hours on this dilemma.

Well, think about it this way ... if you were using another language, you'd have burned 14 hours just getting screens up! At least with Clarion you can spend your time proofing the design and not messing with the details, unless, of course you want to ...

By the way, I saw a post for a Clarion C5b license for $200 ... that would be $200 well spent if I were you. I moved from C4 to C5b the instant I saw it. It is much easier to work with than C4. Plus, you could then take that license and upgrade it further to the latest release. Just a thought.

Greg

Greg,

Thanks for all your help.

Vickie

-- Victoria Galante
831/372-3748
U.S. Cyber-Corps SFS Program
Naval Postgraduate School

Hi Vickie;

Personally, I think your problem is being caused by Referential Integrity ... RIUpdate in other words. Try un-checking it in your dictionary and see what happens.

For my .02 worth, I'm not real sure about your data structure, now that I see what you are doing. Doing a tertiary tree in Clarion is a bit of a challenge. I wanted to do the same thing for a church. Every parent has children, and those children may have children. You not only want to references to the children tied to the parent, but you also want the children to have their own master records, sort of like you've done. I got a little lost when I started trying to account for divorce, which unfortunately happens to Christians, too. What if both parent stay in the church and they both marry other people and both have more kids ... how do you graft the kids onto a new tree, which is what you are doing with the different commands. I never thought of using an alias, though.

Let's take a real tree structure and look at it: a hard drive and its folders is a true tertiary tree, right. Each parent has children except for the lowest levels. Each level has a parent except for the root. When expressed as a data structure its:

root\parent\subfolder ... etc.

Grafting branches is just a matter of changing text strings in essence. This may be more what you want to do. Decide how many levels down you need to go, and provide key fields for each level. Disk space is cheap. I would think it would be much easier to take one command and graft it under a different command with this structure than with the binary tree you've attempted to implement. This is probably the most straight-forward method of creating a hierarchical structure like you're working with, IMHO at least.

Did that make any sense?

Greg

----- Original Message -----
From: gregscales@bitstreet.com
To: Seahorse@Redshift.com
Sent: Thursday, April 03, 2003 9:50 AM
Subject: Re: Tree question

Hi Vickie;

Yes, that's the approach! The reason I specified it was the nature of Clarion. With Clarion "flat" files, it's the best way to express a hierarchical tree. It's not the only way, obviously. A binary tree is expressed like this:

ID | Pointer Up | Left Pointer | Right Pointer

and can be used in memory to store alphabetically sorted lists that can be traversed very rapidly. A tertiary tree could be expressed as:

ID | Pointer Up | 1st Pointer Down | 2nd Pointer Down | 3rd Pointer Down | nth Pointer Down

These type of structures are more appropriate to data that resides in memory than on disk.

I wrote a data base in the 1980's in Basic that was hierarchical in nature but used a Parent File and Child Files (1 to Many relationship). The child file was not keyed like in Clarion, but link listed, like so:

Parent ID | Data Fields | Pointer to Previous Record (0 if First Record) | Pointer to Next Record (0 If Last Record).

This is actually a far superior technique for storing transaction data than Clarion's ISAM (.TPS) format. However, I had to code the entire low level access method myself. The technique would pull up a browse of all transactions using 64K Z80 with floppy drives instead of a hard drive in a couple of seconds. The system supported a grain elevator. A Master file was used to keep track of Grain types and the linked list held the IN / OUT transactions of the elevator.

There are lots and lots of different approaches to data structures. But the one outlined below is probably the best for your particular application, especially written in Clarion.

I'm very honored to help you with your thesis, Vickie. Any further help you need, just let me know.

Greg Scales

Second Approach:

Hi

To do this.

Have all the people in a single table and have a "Parent" field for each record. The topmost record in your tree will not have a parent entry.

There are two ways of displaying and printing this.

1. One way is to have a tree that displays the records, and each sub level is linked to the parent level by the "parent" field. The problem here is you have to judge how many levels you are going to need and create that many aliases.

2. The second way is to have a single list control with a procedure that calls itself over and over again to create an unlimited number of tree levels.

If you need some help with this, you can email me.

Kind Regards

Ben - bdl@riebens.co.za

Hi Ben,

Reviewing your email this morning, I came up with the following (trying to impose an outline format on your assessment):

Have all the people in a single table and have a "Parent" field for each record. The topmost record in your tree will not have a parent entry.

There are two ways of displaying and printing this.

1. One way is to have a tree that displays the records, and each sub level is linked to the parent level by the "parent" field. The problem here is you have to judge how many levels you are going to need and create that many aliases.

2. The second way is to have a single list control with a procedure that calls itself over and over again to create an unlimited number of tree levels.

I may have this indented wrong, but this way, I get one way of organizing the file and two ways to display / print it. When you say two ways of doing it, are you referring to what I have here, or is there another way to organize the file?

Followup question: Is there any way to approach this with a single alias, by doing something like this (call the file OrgFile for simplicity):

- User selects child in OrgFile via browse;
- Access parent in OrgFileAlias, using child's parentKey;
- Redundantly get parent in OrgFile;
- Access grandparent in OrgFileAlias, using parent's parentKey;
- Redundantly get grandparent in OrgFile;
- Etc., until you reach root or desired ancestor.

Does this make any sense at all? Might it work? Does DBMS matter in all this? I kinda like TopSpeed, but am open to whatever, as long as I can use it in C4b.

Thanks,

Vickie Galante

Vickie,

I'm not sure why you need the "alias".

If each record contains a field that is the key value of that individual's superior (or parent, if you will):

1. User selects child in OrgFile via browse;

2. Save Record;

3. Move Parent Key value to OrgFile Key;

4. Get OrgFile,Key;

5. Save Info (if necessary);

6. At Required Ancestor Level?
   Y-Break
   N-Loop to 3

It's still slow, as you say, but is quicker than accessing the file multiple times through the "alias" concept.

From a practical matter, though, it's not really that easy. I used to work for Lockheed-Martin. I was so far down the food chain, that I didn't appear on even low-level org charts. If you wanted to use the above algorithm to find out who was my superior in Florida over my puny self in

Texas, well, it would either take a week of traversing up the chain, or probably not actually yield a result. Lockheed has dozens and dozens of divisions and branches and contracts and ... For a real world app, you'd want to divide the org tree by division or some other convenient method.

Of course, if you use key replication like I was talking about in the previous post, you merely blank out all the fields for every level that you're not interested in, and do one read and voila! You're there. But in the Lockheed example, well, Clarion doesn't support enough key levels to accomplish how far down I was :-)

Greg

From: "info" <info1@email.hinet.hr>
Newsgroups: comp.lang.clarion
Sent: Sunday, February 23, 2003 7:42 AM
Subject: Re: Tree question

Vickie,

If you need something like unlimited level depth, and functions to add level based on the parent/child relations, I have something like that.

Zdravko

From: "r jolda" <rjolda@pdmg.com>
Newsgroups: comp.lang.clarion
Sent: Sunday, February 23, 2003 10:09 AM
Subject: Re: Tree question

Vickie,

The Clarion (Topspeed) Tree uses different files for the different levels within the tree - i.e. the children of a parent come from a different file. If you know how many levels deep you need to go, then you can set this up. However, a more flexible and easier way to do this is to use Paragon D & D Ultratree http://www.paragondandd.com/index.htm - it will be a breeze with this product and you will have much more flexability and will have the ability to jazz it up so that you can impress your thesis committee.

HTH,

Ron Jolda

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix K
Article on Clarion Tree Implementation

A Tree in One File

by David Podger
Published 1997-09-01 in Clarion Magazine - http://www.clarionmag.com/cmag/index.html

I don't know if you have ever felt the need for a file organized as a tree? My wish for one rests on a long-standing desire to find a general solution to the problem of representing a chart of accounts. By 'general' I mean scaleable – simple, shallow structures for small organizations and complex, deep structures for larger ones.

As users of Clarion for Windows, you will be familiar with the Relational Tree template. In this template, each level in the tree is a separate file. The complete tree is defined by a chain of many-to-one relations, which begin with the file at the top of the tree and work down to the bottom. This method has many uses, but because each level is a file, it has a determined number of levels.

A chart of accounts, however, can have any number of levels. It can be shallow in some parts and deep in others. What follows are some ideas for implementing such a tree using a conventional TopSpeed file (such as the *.tps file). Before setting out a possible solution, let's consider some requirements. What follows is my wish list, in no particular order of priority.

Requirements

Browsing the file using its special tree structure to be as simple for the user as any other kind of browse.

1. No practical limit to the number of levels available.
2. Additional, conventional keys on the file are allowed for browsing it by account name, for instance, and for accessing it randomly by name.
3. When browsing by tree, the user is able to collapse and expand levels as in a Relational Tree.
4. The entries at all levels in a tree are dynamic; that is, the user is able to move entries up and down within a given level by pressing an Up or Down button.
5. The levels themselves are dynamic; that is, the user is allowed to promote and demote levels simply by highlighting an entry and pressing a button. A promoted set of records moves up a level, a demoted set moves down one.
6. All of the keys used to order the tree must be conventional keys. One of them must be able to be used randomly to directly locate any record in the file as well as being used for the special purpose of arranging it as a tree. User re-ordering of entries using up/down buttons or promotion/demotion buttons is to have no effect on this key.

7. Coding the tree structure is to require no more than the writing of a purpose-built set of routines within a conventional browse. The main difference between these routines and the conventional ones is that they have a different way of finding the next and previous records.

What follows is not a description of a working solution, it is a sketch of a possible one. I offer it for discussion and further refinement. Perhaps there are undiscovered (by me at any rate) flaws in the suggested solution that make it unworkable. If you are reading this article in Clarion Online then it has already passed the Moseley filter and you may feel it worthwhile to ponder whether it is indeed workable.

Note - While I have examined this article and feel that the concept and idea is technically feasible, I don't filter articles based on my own judgment of the feasibility of the ideas. That said, I did review David's text and was prepared to write a template that compliments this design. Time and resource constraints made this impossible. - Tom Moseley, Publisher

A Possible Solution

Five special fields are required in the records in a file containing a tree:

- Level LONG
- Entry LONG
- Sequence REAL
- PrevLevel LONG
- Collapsed BYTE

Three special keys are constructed, each using two of the above fields:

- Key_LevelEntry - Level and Entry (both ascending)
- Key_LevelSeqAsc - Level and Sequence (both ascending)
- Key_LevelSeqDesc - Level and Sequence (both descending)

The Key_LevelSeq keys are used to construct one part of the order in which a tree browse is presented. Note the word ‘construct’. A tree browse is not presented in the simple order which this key would give to a conventional browse. The key orders records within one level only. The minor field in this key is the one that changes its value in response to the user pressing the Up and Down buttons mentioned above.

The Entry field is used to make links between tree levels. When the last record in a level set has been read using Key_LevelSeqAsc then this field is used to determine where the first record in the next level set is to be found. It is never used to decide the presented order of a browse.

A tree file has a root record with the following values:

- Level zero
- Entry the Level number of the highest level in the tree
- Sequence the Level number of the highest level in the tree

- PrevLevel zero
- Collapsed zero

The root record (found by a SET/NEXT) provides the starting values for a further SET instruction that will get us ready to read the highest level in the file, using Key_LevelSeqAsc. Once this first record in the file has been read, its Entry field permits a SET/NEXT probe (using Key_LevelSeqAsc) to see if it has any child records. If so, the first of these records is the second record in the file to appear in the browse.

A probe for a grandchild record is then done, and, if it is present, it becomes the third record to be displayed. If there are no more grandchildren then the second child record (as per Key_LevelSeqAsc) is the fourth record.

If a record at any level is marked as Collapsed, then this inhibits the probe for its child records.

The diagram in Figure 1 follows the above example exactly. The five values in each box show the values of the five special fields.

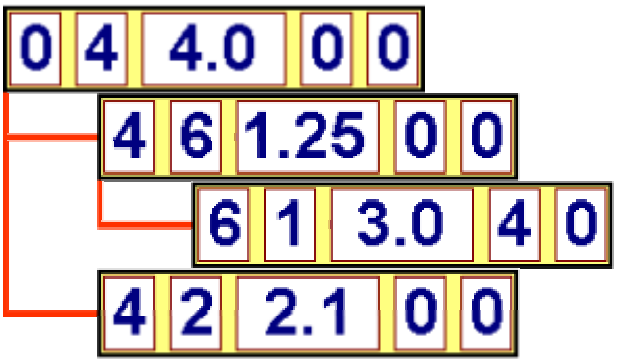


Figure 1: Using the Tree Fields

Root record

The Root values allow a SET/NEXT probe for the first record at the highest level in the tree. This record, in turn, allows a probe for a grandchild record, whose Level field content is given by the Entry field in the level above. When there are no more grand-children, the next record in the child sequence is read.

The above process is recursive - the same simple logic is used repetitively to proceed down the tree, no matter what level that logic is dealing with.

The Five Fields - their usage explained

The value in an Entry field never changes during the lifetime of a record. The Entry field is unique. Other files can hold its value and always recover a given record at random from a tree file. Subsequent movement forwards or backwards in the tree file can in principle begin from any such random beginning point.

In all but one instance, the values in Level and PrevLevel are also unchanging. That instance is when a level promotion or demotion takes place.

The PrevLevel field allows the writing of a special-purpose PreviousRecord routine. With it, the routine can find its way backwards, all the way to the first record in the tree. In effect, the file is a doubly-linked list, with the Entry field serving the purpose of the forward link whilst PrevLevel provides the backwards link.

When a tree file is first loaded with data the Level and Entry fields are allocated in a simple sequence starting at zero (0) for the root record and working upwards one at a time. So it makes sense that, if some part of the file has a preferred order (say alphabetic by name), then it should be loaded in that order in the first instance. On the first load, all Sequence fields are set to the value of their respective Entry fields.

The Sequence field is REAL to allow arithmetic on the field when the Up or Down button is pressed. To move a record above its neighbor, first add together its Sequence field and that of its neighbor's neighbor. Divide the result by two and write this back into the record's Sequence field. The special cases that arise at the top and bottom of a level are easily dealt with. The important point is that the above simple method works regardless of the level being re-ordered by changes to the Sequence field.

The Collapsed field is a TRUE/FALSE indicator that all the children under a given record have been collapsed and are not to be visible in a browse.

I should note that this article does not consider the possibility of range and filter limits. Filtering is fraught with difficulty since filtered records might, by their absence, break the link between records. A range limit that allowed only one branch of a tree to be shown is, however, a possibility.

Climbing back up the Tree

This example shows how going backwards up a tree uses the Key_LevelSeqDesc key.

```
Root record
 A level record*
 B level record
  C level record
  C level record
  C level record
 B level record
  C level record
  C level record**
 A level record
 B level record
```

Let us suppose that we are positioned on the **second A level record** and we execute the PreviousRecord routine. If we are going backwards in a fully expanded tree, we will want it to find the **last C level record** (marked with a double asterisk). The steps are as follows:

1. A SET/NEXT on Key_LevelSeqDesc finds the preceding A level record
2. Using the Entry field in this record, we find that it has child (B) records
3. Using Key_LevelSeqDesc we retrieve the last B level record
4. Using the Entry field in this record, we find that it has child (C) records
5. Using Key_LevelSeqDesc we retrieve the last level C record
6. It has no child records. We have found the correct previous record.
7. We can then repeat this recursive process as we move up another record. It is much simpler this time:
8. A SET/ NEXT on Key_LevelSeqDesc finds the preceding C level record
9. It has no child records. We have found the correct previous record.
10. Now, suppose that the A level record marked with a single asterisk is in fact Collapsed. Then, Step 1. above will read that record, determine that it is collapsed, and not do a probe for child records. The correct previous record has been found in one step.

Promotion and Demotion

We will use the same diagram from above to illustrate promotion and demotion:

```
Root record
 A level record
 B level record
  C level record
  C level record
  C level record
 B level record *
  C level record *
  C level record *
 A level record
 B level record
```

Let us say the user wishes to promote the three asterisked records. The B record is to become an A and its C records are to become B's. The user does this by highlighting the B level record and pressing the Promote button. The steps that follow are:

1. Find the PrevLevel of the A level record (B's parent). In this case it is the root record (zero). This value is needed when we change the PrevLevel field in the promoted B record.
2. Change the PrevLevel and Level fields in the promoted B record. In this case, PrevLevel becomes zero and Level takes on the value of the A record's Level field. No change is made to Entry or Sequence.
3. Proceed to change all the children, grandchildren etc. of the promoted record in the same manner.
4. Promotion of all children applies regardless of the setting of the Collapsed byte.

Demotion of the above boxed records would mean demoting the highlighted B record to become the last in the list of C records immediately above it and demoting its child records to become D records. There are obvious restrictions. The last record in the diagram (a B record) cannot be demoted as it has nowhere to go. Without spelling out a detailed procedure, it is clear that demotion is as easy to do as promotion.

Inserting and Deleting records

Insertion of a record occurs at the level of the highlighted record and immediately below it. To insert the first of a set of child records below a highlighted parent, first insert it at the parent's level and then demote it. Keep the highlight bar on the just-demoted record and the next insertion will be at the same (i.e. child) level. The Entry field must receive the next highest value available.

Deletion of a record with children has to follow the rules set in the browse; that is, it is either restricted or cascaded. Since all the relations between records occur within one file, it stands to reason that a template for a tree browse would allow for this choice.

Conclusion

Failing the discovery of a fatal objection to the above method, a tree of any practical number of levels can be held within a *.tps file (or in any of the comparable files with Clarion drivers). The extra overhead caused by the special purpose routines needed to browse and maintain the tree should not be excessive. The extra fields needed in every record do not occupy that much space.

I have given no consideration to multi-user issues. Promotion and demotion, either of which involves cascading a change through an unknown number of child records, would also require file locking. Changing the relative order of a record would entail locking only that record (and perhaps two adjacent ones). Cascading deletions may also require file locking.

Note - It was originally the intent of Clarion Online to have an article detailing the construction of a template to compliment this specification accompany this article. This turned out not to be feasible because of the complexity of the design, and Mr. Podger's desire that the template be a page-loaded affair, like the BrowseBox. - Publisher

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Ernest McDuffie
   National Science Foundation
   Arlington, VA

4. RADM Zelebor
   N6/Deputy DON CIO
   Arlington, VA

5. Russell Jones
   N641
   Arlington, VA

6. David Wirth
   N641
   Arlington, VA

7. CAPT Sheila McCoy
   Headquarters U.S. Navy
   Arlington, VA

8. CAPT Robert Zellmann
   CNO Staff N614
   Arlington, VA

9. Dr. Ralph Wachter
   ONR
   Arlington, VA

10. Dr. Frank Deckelman
    ONR
    Arlington, VA

11. Richard Hale
    DISA
    Falls Church, VA

12. George Bieber
OSD
Washington, DC

13. Deborah Cooper
DC Associates, LLC
Roslyn, VA

14. David Ladd
Microsoft Corporation
Redmond, WA

15. Marshall Potter
Federal Aviation Administration
Washington, DC

16. Ernest Lucier
Federal Aviation Administration
Washington, DC

17. Keith Schwalm
DHS
Washington, DC

18. RADM Joseph Burns
Fort George Meade, MD

19. Howard Andrews
CFFC
Norfolk, VA

20. Steve LaFountain
NSA
Fort Meade, MD

21. Penny Lehtola
NSA
Fort Meade, MD

22. LT Luciana Sung
USN – FIWC
Virginia Beach, Virginia

23. LT Andrew Lamorie
USN – FIWC
Virginia Beach, Virginia

24. Dr. Thomas Otani, Thesis Advisor
NPS
Monterey, CA

25. J.D. Fulp, Thesis Second Reader
NPS
Monterey, CA

26. Victoria Galante
Civilian, Naval Postgraduate School
Monterey, CA